

AD-A036 440

FRANKFORD ARSENAL PHILADELPHIA PA

F/6 19/5

INVESTIGATION OF KALMAN FILTER DESIGNS FOR AIR DEFENSE BASED ON--ETC(1)

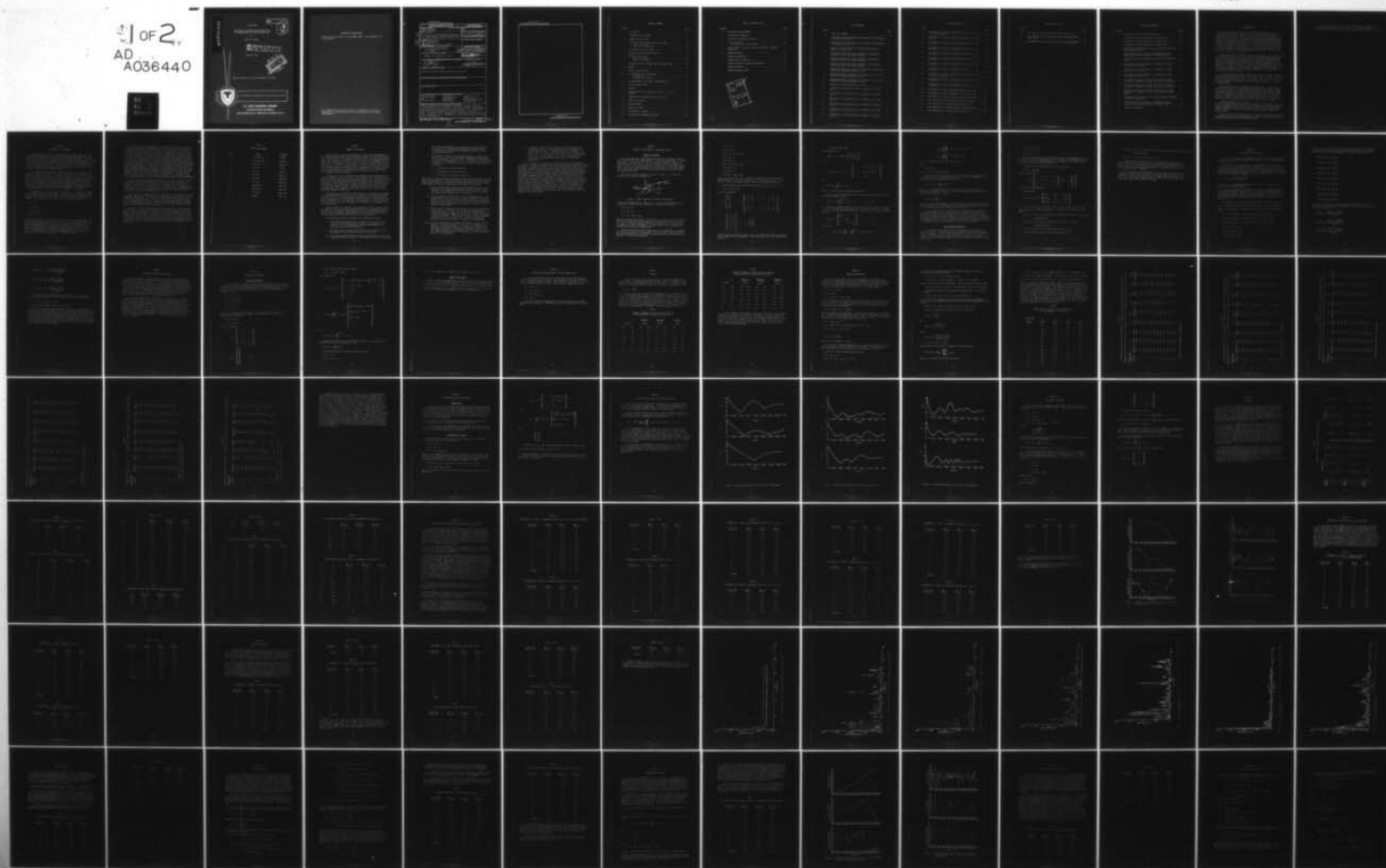
SEP 76 W J DZIWA, L VITALE, M MINTZ

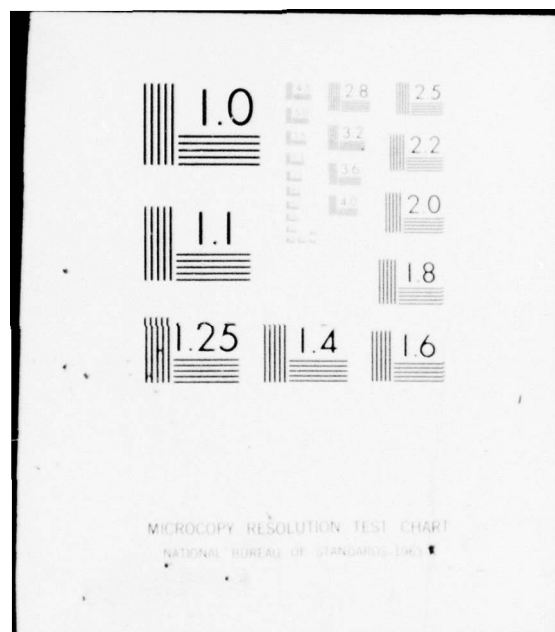
UNCLASSIFIED

FA-TR-76058

NL

1 OF 2
AD
A036440





ADA036440

AD

FA-TR-76058

INVESTIGATION OF KALMAN FILTER DESIGNS FOR
AIR DEFENSE BASED ON ATTACK AIRCRAFT DATA

by

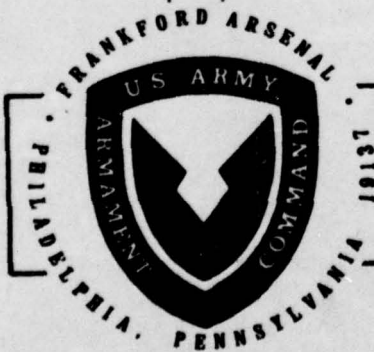
Walter J. Dziwak

COPY AVAILABLE TO DDC DOES NOT
PERMIT FULLY LEGIBLE PRODUCTION

September 1976



Approved for public release; distribution unlimited



Fire Control Development and Engineering Directorate

U.S. ARMY ARMAMENT COMMAND
FRANKFORD ARSENAL
PHILADELPHIA, PENNSYLVANIA 19137

DISPOSITION INSTRUCTIONS

Destroy this report when it is no longer needed. Do not return it to the originator.

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

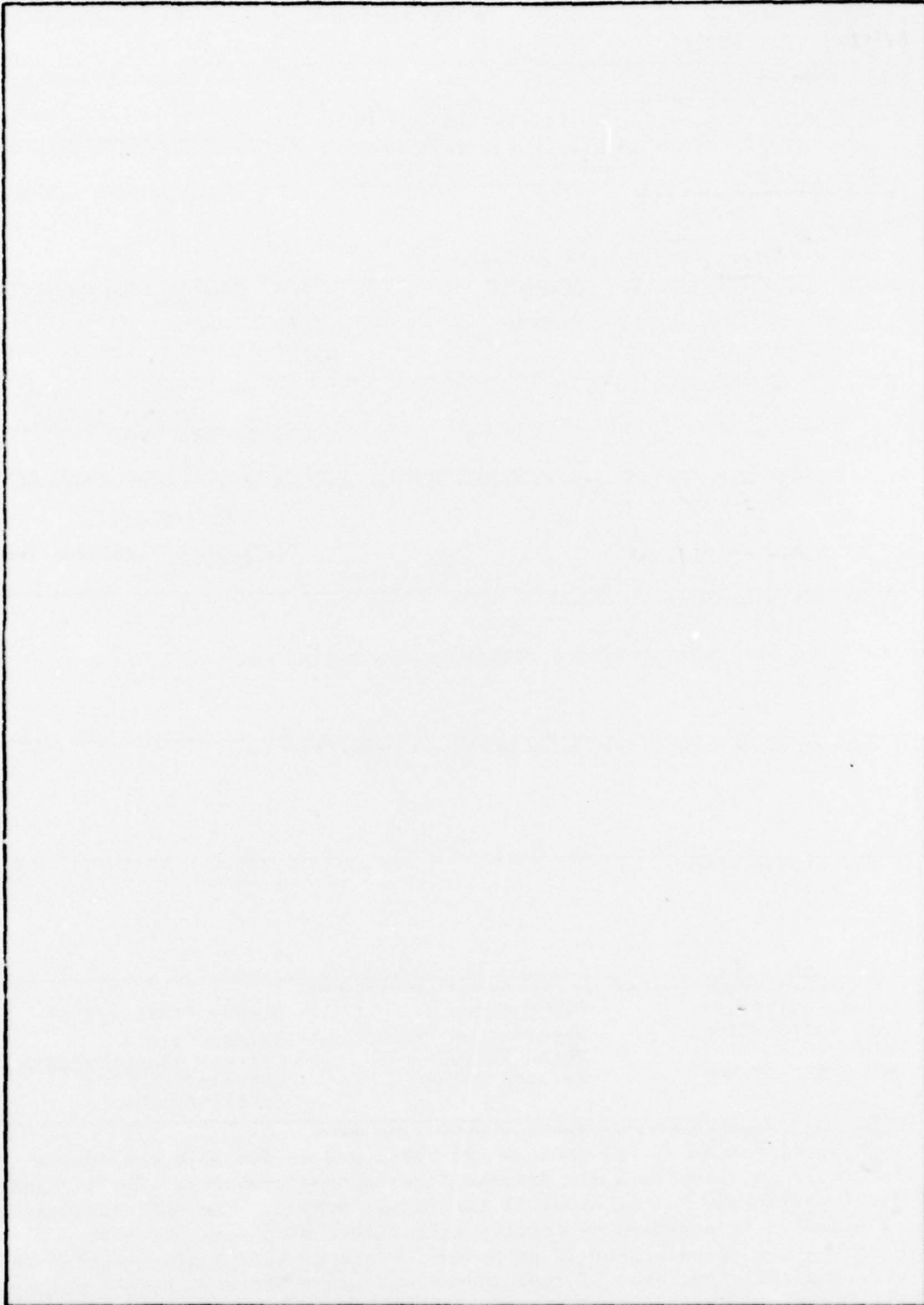
UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM															
1. REPORT NUMBER 14 FA-TR-76058	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER															
4. TITLE (and Subtitle) Investigation of Kalman Filter Designs for Air Defense Based on Attack Aircraft Data		5. TYPE OF REPORT & PERIOD COVERED Final Engineering Report															
6. AUTHOR(s) Walter J. Dziwak, Larry Vitale Max Mintz, University of Penna.		7. PERFORMING ORG. REPORT NUMBER															
8. CONTRACT OR GRANT NUMBER(s)																	
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Defense Weapons Branch FCW-D(N6200)		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS AMCMS CODE: 662603.11.H7800 DA PROJECT: 1W662603AH78															
11. CONTROLLING OFFICE NAME AND ADDRESS ARMCOM, Rock Island, Ill 61201		12. REPORT DATE September 1976															
13. NUMBER OF PAGES 146		14. SECURITY CLASS. (of this report) UNCLASSIFIED															
15. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) 12 146p.		16. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A															
17. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.																	
18. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)																	
19. SUPPLEMENTARY NOTES																	
20. KEY WORDS (Continue on reverse side if necessary and identify by block number) <table border="0"> <tr> <td>Kalman Filtering</td> <td>Plant Model</td> <td>Steady State Filter</td> </tr> <tr> <td>Auto Correlation</td> <td>Observation Model</td> <td>Kalman Gain</td> </tr> <tr> <td>FACT</td> <td>White Noise</td> <td>Process Identification</td> </tr> <tr> <td>Nth Order Markov</td> <td>Colored Noise</td> <td>Correlation Time</td> </tr> <tr> <td></td> <td></td> <td>Settling Time</td> </tr> </table>			Kalman Filtering	Plant Model	Steady State Filter	Auto Correlation	Observation Model	Kalman Gain	FACT	White Noise	Process Identification	Nth Order Markov	Colored Noise	Correlation Time			Settling Time
Kalman Filtering	Plant Model	Steady State Filter															
Auto Correlation	Observation Model	Kalman Gain															
FACT	White Noise	Process Identification															
Nth Order Markov	Colored Noise	Correlation Time															
		Settling Time															
21. ABSTRACT (Continue on reverse side if necessary and identify by block number) <p>Several Kalman filter designs are developed as possible candidates for implementation in a air defense fire control computer. The designs differ primarily in the model of the target motion. The effectiveness of each design is assessed by driving each filter with real aircraft data. A candidate filter is proposed for air defense implementation which offers simplicity, ease of implementation, and relatively high accuracy in the state estimates.</p>																	

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

TABLE OF CONTENTS

SECTION	PAGE
INTRODUCTION	7
1 STATEMENT OF THE PROBLEM	9
2 SUMMARY AND CONCLUSIONS.	12
3 PROTOTYPE FILTER DESIGN - NINE STATE FILTER.	15
Model of the Plant.	15
Model of the Observation.	18
4 INITIALIZATION OF THE FILTER	21
5 SUBOPTIMAL FILTER DESIGN PHILOSOPHY.	24
6 THE FIVE STATE FILTER.	25
Model of the Plant.	25
Model of the Sensor	27
7 MODIFIED FIVE STATE FILTER - THE FOUR STATE FILTER	28
8 RESULTS.	29
9 PROCESS IDENTIFICATION	31
10 THE THREE AND FOUR STATE FILTERS	41
Introduction.	41
The Three State Filter.	41
11 AUTOCORRELATION OF THE FACT - ACCELERATION DATA.	43
12 THE FOUR STATE FILTER.	47
13 RESULTS.	49
14 OPTIMIZATION OF THE THREE STATE FILTER FOR ALL FLIGHT PASSES	55
15 COMPARISON OF THE THREE AND FOUR STATE FILTERS	64
16 STEADY STATE FILTER.	67
17 IMPROVED SENSOR.	79
18 RATE AIDED FILTER.	81
19 CORRELATED NOISE SENSOR.	85
20 SIMULATION OF THE NINE STATE FILTER.	89

TABLE OF CONTENTS (Cont)

APPENDIX	PAGE
A THE KALMAN FILTER EQUATIONS.	91
B DISCRETE PLANT EQUATIONS	92
C ACCELERATION CORRELATION FUNCTIONS	95
D SENSOR COVARIANCE.	98
Approximations to the <u>R</u> Matrix.	102
E INITIALIZATION OF POSITION - VELOCITY, VELOCITY - VELOCITY TERMS OF P_0	104
F COMPUTER SOFTWARE.	106
NARRATIVE DESCRIPTION.	108
PROGRAM FLOW AND ANALYSIS.	110
OPTIMAL FILTERING AND SENSING ALGORITHMS	124
INDEX TO ROUTINES.	131
COMPUTER PROGRAM LISTING	134

ACCESSION for	
RTIS	White Section <input checked="" type="checkbox"/>
BBC	Bull Section <input type="checkbox"/>
UNANNOUNCED	
JUSTIFICATION	
BY DISTRIBUTION/AVAILABILITY CODES	
Dist.	AVAIL. and/or SPECIAL
A	

LIST OF TABLES

TABLE		PAGE
1	Flight Data Summary.	11
2	Comparison Between the Fourth and Fifth Order Filters Along the y Coordinate for Pass No. 1	29
3	Comparison Between the Fourth and Fifth Order Filters Along the y Coordinate for Pass No. 9	30
4	Number of Data Points for each Flight Path and for each segment of a Flight Path	33
5	Theoretical Values of "a" for x Component of Acceleration Modeled as a First Order Markov Process.	34
6	Theoretical Values of "a" for y Component of Acceleration Modeled as a First Order Markov Process.	35
7	Theoretical Values of "a" for z Component of Acceleration Modeled as a First Order Markov Process.	36
8	Theoretical Values of ω_0 and β for x Component of Acceleration Modeled as a Second Order Markov Process	37
9	Theoretical Values of ω_0 and β for y Component of Acceleration Modeled as a Second Order Markov Process	38
10	Theoretical Values of ω_0 and β for z Component of Acceleration Modeled as a Second Order Markov Process	39
11	Comparison of Three and Four State Filters	50
12	Optimization of Three State Filter (y Component) for Flight Pass 1	51
13	Optimization of Four State Filter (y Component) for Flight Pass 1	51
14	Optimization of Three State Filter (y Component) for Flight Pass 9	52
15	Optimization of Four State Filter (y Component) for Flight Pass 9	53
16	Optimization of Three State Filter (y Component) for Flight Pass 12.	54
17	Optimization of Four State Filter (y Component) for Flight Pass 12.	54

LIST OF TABLES (Cont)

TABLE		PAGE
18	Performance of 3 State, y Coordinate Filter for "a" = .05, <u>R</u> = State Dependent.	56
19	Performance of 3 State, y Coordinate Filter for "a" = .05, <u>R</u> = 31	56
20	Performance of 3 State, y Coordinate Filter for "a" = .05, <u>R</u> = 100.	57
21	Performance of 3 State, y Coordinate Filter for "a" = .1, <u>R</u> = 100.	58
22	Performance of 3 State, x Coordinate Filter for "a" = .1, <u>R</u> = 100.	58
23	Performance of 3 State, z Coordinate Filter for "a" = .1, <u>R</u> = 100.	59
24	Performance of 3 State, z Coordinate Filter for "a" = .05, <u>R</u> = 100.	60
25	Performance of 3 State, z Coordinate Filter for "a" = .1, <u>R</u> = 100.	60
26	Performance of 4 State, y Coordinate Filter for $\omega_0 = .1$, $\beta =$.03, <u>R</u> State Dependent	64
27	Performance of 4 State, y Coordinate Filter for $\omega_0 = .1$, $\beta =$.03, <u>R</u> = 31.	65
28	Performance of 4 State, y Coordinate Filter for $\omega_0 = .1$, $\beta =$.03, <u>R</u> = 100	65
29	Performance of 3 State, x Coordinate Steady State Filter	67
30	Performance of 3 State, y Coordinate Steady State Filter	68
31	Performance of 3 State, z Coordinate Steady State Filters. . . .	69
32	Total RMS Errors for the 3 State, Steady State Filters	69
33	Total RMS Errors for 3 State Filter with <u>R</u> = 100	70
34	Total RMS Errors for 3 State Filter with Improved Sensor	79
35	Total RMS Errors for 3 State, Rate Aided Filters	83

LIST OF TABLES (Cont)

TABLE		PAGE
36	Total RMS Errors for 3 State, Rate Aided Steady State.	84
37	Total RMS Errors for 3 State Filters with Coorelated Sensor Noise Inputs	86
38	Total RMS Errors for 9 State Filter with <u>R</u> , State Dependent. . .	89

LIST OF ILLUSTRATIONS

FIGURE		PAGE
1	Sensor Variables in Inertial Coordinate Set.	15
2	Acceleration Correlation Functions for Flight Pass 1	44
3	Acceleration Correlation Functions for Flight Pass 9	45
4	Acceleration Correlation Function for Flight Pass 12	46
5	Filtered and True Position, Velocity and Acceleration Profiles Along x for Flight Pass 1.	62
6	State Estimation Error Curves for x Component of Target Position, Velocity and Acceleration for Flight Pass 1.	63
7	Position, Velocity and Acceleration Gains for the Three State Filter	72
8	Error Curve for State Estimates of y Component of Target Position for Flight Pass 1	73
9	Error Curve for State Estimates of y Component of Target Velocity for Flight Pass 1	74
10	Error Curve for State Estimates of y Component of Target Acceleration for Flight Pass 1	75
11	Error Curve Generated with Steady State Filter for State Estimates of y Component of Target Position for Flight Pass 1. . .	76
12	Error Curve Generated with Steady State Filter for State Estimates of y Component of Target Velocity for Flight Pass 1. . .	77
13	Error Curve Generated with Steady State Filter for State Estimates of y Component of Target Acceleration for Flight Pass 1	78
14	Filtered and True Position, Velocity and Acceleration Profiles Along y for Flight Pass 1.	87
15	State Estimation Error Curves for y Component of Target Position, Velocity and Acceleration for Flight Pass 1.	88
C-1	Contour of Integration	97

INTRODUCTION

The operation of a fire control system entails several tasks. The weapon system must acquire a target, lock on and maintain track, accurately estimate the present target state, and provide gun orders based upon the predicted future position of the target. The target state may consist of any convenient set of variables used to describe the dynamics of the target motion. Typically, these are the target position, velocity, and acceleration, as well as auxiliary variables which describe the stochastic behavior of the motion. Accuracy in estimation of target state is a necessary condition for predicting its future position. Unfortunately, it is not a sufficient condition, since the target can undergo unexpected maneuvers during the time-of-flight of the projectile resulting in a miss. The purpose of this report is to focus on the problem of target state estimation.

Modern state estimation theory provides a powerful tool for estimating target state from sensor data of target position and rate information, if available. The power of this theory, sometimes referred to as Kalman filtering, derives from the fact that the estimation procedure is iterative so that only the present sensor measurements and the previous "optimal" target state estimates are needed to compute the "optimal" estimate of the present target state. This feature of the theory obviates the need to store a large number of data points. Furthermore, the solution for the "optimal" estimate is related linearly to the observations and the previous optimal estimate which is an additional simplifying feature of the theory.

This report describes several Kalman filter designs, from which a candidate for possible implementation in a fire control system is chosen. In selecting the final filter design, two criteria are applied. One is that the number of state variables be relatively small so as to reduce the computational time for the state estimates. The second is that the relative accuracy of the state estimates be high.

The lack at present of a complete model for maneuvering targets as well as insufficient data on sensor performance characteristics generally imposes a limit on the accuracy of filter design and hence, its performance. Computer requirements impose an additional constraint in that an accurate target model, which may require for its description a large number of state variables, may result in a relatively long time to solution for the estimates of the state variables. Finally, the statistics governing real aircraft motion may not, in fact, be Gaussian as required by the theory to achieve global optimality.

Although proposals for using Kalman filters in air defense date back to the 1960's, the filter design presented here is unique for two reasons. Compared to designs proposed heretofore, the filter described here is of minimal mathematical complexity. Secondly, flight data on high performance aircraft made recently available, is utilized to optimize the filter design through model identification.

A brief description of the available flight path data upon which the design is based as well as the underlying assumptions leading to a particular design philosophy are discussed in the following section.

It is assumed that the reader is familiar with Kalman filtering theory. For the reader's convenience, a brief summary of the Kalman equations is provided in Appendix A, together with the underlying assumptions of the theory.

SECTION 1

STATEMENT OF THE PROBLEM

During the summer of 1974, the Frankford Arsenal Capabilities Test (FACT) was conducted at the Naval Weapons Center at China Lake, California. In these tests, tactical aircraft were flown to simulate combat conditions. Accelerometers placed on board the aircraft recorded aircraft accelerations for 20 flight paths while a ground based Nike Hercules radar recorded aircraft position. Forty more simulated attack missions were flown for which only radar derived target position information was recorded. At the completion of these tests, it was recognized that the FACT data may provide the basis for a filter design. The key design goals would be accuracy and simplicity.

In designing a suitable filter, one must decide upon a coordinate system in which to filter sensor data. The most likely candidates are rectangular and spherical coordinates. Spherical coordinates provide a natural set in which to operate the filter because the sensor data is already in spherical form. However, one is faced with the problem of adequately modeling the target motion in this set. The restrictions on the model of target dynamics arise from the Kalman theory which states that the plant equations (a name given to the equations describing the target motion) must be linear differential equations in the state variables. If the problem involves non-linear dynamics, then the Kalman theory is still applicable, provided that suitable linearization is possible.

The difficulty in suitably modeling target dynamics in spherical coordinates is illustrated by the following example. Suppose that x_1 is the state variable for the range to the target, designated by R in spherical coordinates. If one models the acceleration in x_1 as composed of both a random and a deterministic component x_3 and x_4 , respectively, then a four state model may look like

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = x_3 + x_4$$

$$\dot{x}_3 = \alpha x_3 + \beta u$$

$$\dot{x}_4 = 0$$

with u = white noise. Consider, for the sake of simplicity, the deterministic portion of the model, obtained by setting $\alpha = \beta = 0$. The resulting expressions can then be easily solved for x_1 to yield a 3rd degree polynomial in time. However, under almost all circumstances, the range R is an infinite series in time. Strictly speaking, then, one would require an infinite set of state variables to accurately describe the deterministic portion of the acceleration in R . One may, nevertheless, proceed to implement such a model under the assumption that the state estimates will be reasonable anyway.

By filtering in the rectangular set, this problem is obviated. However, difficulty arises when one attempts to describe the sensor by a linear model (linear in the state variables) in this set. The source of this difficulty is the fact that the sensor data is in spherical form whereas the state variables are now in described rectangular coordinates. Conversion of the sensor data into the rectangular set is a non-linear transformation which results in equations which are non-linear rather than linear in the state variables as required by the theory. This, however, is a minor drawback for two reasons. The first is that suitable linearization of the equations describing the sensor model is possible but with the slight penalty that the coefficients of the state variables become time dependent. The second reason is that more is generally known about sensor error characteristics than about aircraft dynamics. One therefore, has more information available to construct a model representative of a real sensor than information about aircraft dynamics to construct a single model which adequately describes the spectrum of real aircraft maneuvers. For this reason, emphasis in this report will be placed on the plant model. The sensor noise will be modeled as a white noise sequence, although filter performance against sensor errors described by a first order Markov process will also be investigated.

Another important consideration in choosing a suitable coordinate frame is the description of the stochastic behavior of the aircraft acceleration. If the target acceleration is characterized by stationary Gaussian statistics, a debateable point at this time, then Gaussian stationarity is more likely to be manifest in the inertial frame (rectangular frame) than in a rotating non-inertial polar frame. For the reasons mentioned above, inertial (fixed to the earth) coordinates are chosen for modeling the filter.

After choosing a suitable coordinate system, one must adequately model both the dynamic and the stochastic aspects of target motion. Two classes of target models are investigated. In one class, the target acceleration is modeled as a first order Markov process; in the other class, the acceleration is modeled as a second order Markov process. Based on their performance against the FACT data, a recommendation is made as to the filter design that should be incorporated into an air defense fire control system.

All tests of filter performance are done by exercising the filter against the FACT data. This data, which provides target position, velocity and acceleration in cartesian coordinates is available in 1/10 sec increments. Thus, the sampling rate used throughout the report is 10 data points per second. Furthermore, of the 20 flight paths for which position, velocity and acceleration data was obtained, 12 are useable. A classification of these useable flight paths is provided in Table 1. For more details concerning FACT, the reader is referred to the report, "The Frankford Arsenal Capabilities Test" (FACT) July 1974 FA-TR-74001.

TABLE 1
FLIGHT DATA SUMMARY

<u>Pass</u>	<u>Path</u>	<u>Ordinance</u>
1	45° Dive Toss	3 Bombs
2	30° Dive Toss	Dummy Run
3	45° Dive Toss	Dummy Run
4	45° Dive	3 Bombs
5	30° Dive	Dummy Run
6	30° Dive	Dummy Run
7	30° Dive	Dummy Run
8	45° Dive	Dummy Run
9	Pop-up/dive	Dummy Run
10	Pop-up/dive	Dummy Run
11	Pop-up/dive	Dummy Run
12	Laydown	Dummy Run

SECTION 2

SUMMARY & CONCLUSIONS

Several filter designs were investigated for possible implementation in an air defense weapon system. The performance of each design was determined by the total RMS error in the estimates of target position, velocity, and acceleration. Each design was "optimized" in the sense that the free parameters of the design were adjusted so as to yield the most accurate state estimates (lowest RMS errors). The targets for which these filters were designed were not simulated but real. That is, data in the form of position, velocity, and acceleration time histories of real aircraft flying simulated combat missions was used to aid in the selection of the filter design parameters. This data, called FACT (Frankford Arsenal Capabilities Test), provided inputs into the filter via the weapon system sensor model.

In addition to real aircraft data, which permitted the "optimization" of filter performance, a simplification of the usual nine state filter design was sought in order to minimize the computational burden of the fire control computer. This simplification was successfully achieved by decoupling the nine state filters with little degradation in filter performance. Each three state filter incorporates a target state model in which the target acceleration is modeled as a first order Markov process. Although decoupled filters involving more state variables were designed, it was found that they did not offer improved performance against the FACT data.

A decoupled four state filter in which the target acceleration was modeled as a second order Markov process was also designed. This design was purposely constructed in such a way as to produce an acceleration correlation function that is represented as an exponentially decaying sinusoid. Although the target model incorporated into this four state filter appeared to be more realistic, the performance of the filter nevertheless did not prove to be superior to the three state design.

In most simulations involving filter performance, the fire control sensor was modeled to provide position information only with a standard deviation in the range error of 10m and a standard deviation in the azimuth and elevation errors of 1m. Nevertheless, the performance of rate aided filters as well as filters using more accurate sensor inputs was also investigated.

The conclusions reached in this report are summarized as follows:

- i) The three filters which require no matrix inversion for the solution for the state estimates in rectangular coordinates offer a simple filter design and adequate target state estimates (i.e. - position, velocity, and acceleration).
- ii) The three state filters are characterized by a 2 second settling time against targets undergoing constant $4g$ maneuvers.
- iii) For a sensor providing position data only and characterized by a 10m standard deviation in range and a 1m standard deviation in both azimuth

and elevation measurements, the average RMS position measurement errors over all the flight passes contained in FACT is 11.3m. The three state filters reduce the RMS position error by 37% to an average value of 7.1m.

- iv) The three state filters do not perform as well as a single nine state filter. This is to be expected. Nevertheless, the degradation in performance is minimal. Over the ensemble of flight paths considered (12 in number), the decrease in the RMS error of the state estimates for the nine state filter over the three state filter is as follows:

- a) 2.7% decrease in position RMS error
- b) 10.9% decrease in velocity RMS error
- c) 16.4% decrease in acceleration RMS error

The 10.9% and 16.4% figures are of little practical significance when the actual RMS values are compared. For the velocity RMS, the nine state filter yields a value of 13.1m/sec vs. 14.7m/sec for the three state filters and for the acceleration RMS, the figure are, respectively, 15.3m/sec² and 18.3m/sec².

- v) The three state steady state filters (fixed gain filters) perform almost as well as the non-steady state three state filters. The difference in performance is negligible. The storage requirements and computational complexity are both minimal.
- vi) If the accuracy of a position only sensor is doubled, that is, the standard deviations of the range, azimuth and elevation errors are reduced from 10m, 1m, 1m to 5m and 1/2m 1/2m respectively, then the position state estimates improve by a factor of 1.9 to 1 whereas the velocity and acceleration estimates improve by factors of 3 to 2 and 6 to 5 respectively.
- vii) Addition of rate measurements (the accuracy of position measurements remains fixed at σ values of 10m, 1m, and 1m) characterized by standard deviations in range rate, azimuth and elevation rates of 5 m/sec and 1/2m/sec, 1/2m/sec results in the following improvements in the state estimates. The position estimates improve in accuracy by a factor of 2.3 to 1, whereas the velocity and acceleration estimates improve by factors of 4 to 1 and 9 to 5 respectively.
- viii) In view of vi) and vii), it appears that addition of rate measurements (of the relative quality given in item vii) improves the accuracy of the state estimates to a greater extent than does the doubling of the accuracy of position only sensors. The position-rate sensor improves filter performance over the improved position only sensor by a factor of 6 to 5 in position, 2.7 to 1 in velocity, and 3 to 2 in acceleration.

- ix) In items i) through vii), the sensor noise was modeled as being uncorrelated in time. If the sensor model, as it appears in the filter equations, remains unchanged but the sensor data which is inputted into the filter is time correlated with a correlation time constant of .1 sec., then the state estimates improve significantly. Specifically, the RMS position errors drop by 53% whereas the velocity and acceleration RMS errors decrease by 42% and 19% respectively.

A brief incursion into the area of process identification was made for the purpose of getting some handle on the approximate values of the filter design parameters. The most significant result of this brief study, however, was that the target acceleration could not be statistically characterized as being a stationary process. Whether or not it is Gaussian remains in doubt although no firm conclusion in this regard could be made with the data that was available. Furthermore, the statistical behavior of the acceleration process appeared to be strongly dependent upon the target attack mode. This information, it was recognized, could be useful in the design of adaptive filters. A thorough study of the available flight data using autoregressive models will undoubtedly provide some very useful information which is needed for the purpose of obtaining good statistical models of target motion. Such models would provide not only a means of further enhancing the quality of target state estimates, but would also provide a means of better predicting the future position of the target, which is ultimately what is desired. The only requirement is that plenty of data, representative of the threat posed to air defense systems, be available for further analysis.

SECTION 3

PROROTYPE FILTER DESIGN - NINE STATE FILTER

MODEL OF THE PLANT

The filter described in this section has nine state variables, three for each of the components of target position, velocity and acceleration. Although this filter is not proposed for use in an actual air defense system, it is described here for two reasons. First, it provides a logical framework from which a class of lower order filters can be constructed. Second, the proposed filter design will be tested against this nine state filter. One will then be able to judge the degradation in filter performance when certain simplifying approximations are made.

The target position variables are labeled in Figure 1. We model each component of the target acceleration as

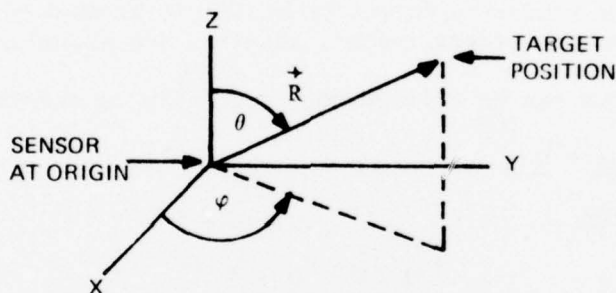


Figure 1. Sensor Variables in Inertial Coordinate Set

a first order Markov process. Thus, if $x_1 = x$ is the state variable for the x component of target position, we model the target motion along X as

$$(3-1) \quad \dot{x}_1 = x_2$$

$$(3-2) \quad \dot{x}_2 = x_3$$

$$(3-3) \quad \dot{x}_3 = -a_1 x_3 + b_x u_x$$

where u_x represents white noise. The quantity a_1 is related to the correlation time for \ddot{x} the acceleration process and b_x is the intensity of the white noise. It is best to obtain a numerical value for a_1 experimentally in such a way as to optimize the filter performance. A suitable value of a_1 can, however, be obtained from theoretical considerations.

The equations describing the target motion in the remaining two components are identical in form to the above model for motion along X . Thus, there are nine state variables and nine linear differential equations describing the target motion. The state equations are

$$(3-4) \quad \dot{x}_1 = x_2$$

$$(3-5) \quad \dot{x}_2 = x_3$$

$$(3-6) \quad \dot{x}_3 = -a_1 x_3 + b_x u_x$$

$$(3-7) \quad \dot{x}_4 = x_5$$

$$(3-8) \quad \dot{x}_5 = x_6$$

$$(3-9) \quad \dot{x}_6 = -a_2 x_6 + b_y u_y$$

$$(3-10) \quad \dot{x}_7 = x_8$$

$$(3-11) \quad \dot{x}_8 = x_9$$

$$(3-12) \quad \dot{x}_9 = -a_3 x_9 + b_z u_z$$

where x_4 and x_7 are the y and z components of target position respectively. Because there is no preferred direction in the horizontal plane, $a_1 = a_2$ and $b_x = b_y$. There is no a-priori reason, however, for assuming that $a_3 = a_1$.

These equations can be written more concisely in matrix form as

$$(3-13) \quad \dot{\underline{x}} = \underline{F} \underline{x} + \underline{G} \underline{u}$$

where, by inspection

$$\underline{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_9 \end{bmatrix}, \quad \underline{F} = \begin{bmatrix} 0 & 1 & 0 & \cdot & \cdot & \cdot & \cdot & 0 & 0 & 0 \\ 0 & 0 & 1 & \cdot & \cdot & \cdot & \cdot & 0 & 0 & 0 \\ 0 & 0 & -a_1 & \cdot & \cdot & \cdot & \cdot & 0 & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & \cdot & \cdot & \cdot & \cdot & 0 & 1 & 0 \\ 0 & 0 & 0 & \cdot & \cdot & \cdot & \cdot & 0 & 0 & 1 \\ 0 & 0 & 0 & \cdot & \cdot & \cdot & \cdot & 0 & 0 & -a_3 \end{bmatrix}$$

$$\underline{G} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ b_x & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & b_y & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & b_z \end{bmatrix}, \quad \underline{u} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}$$

Because sensor data comes in discrete form, the discretized form of the Kalman filter equation will be implemented. Once discretized, the plant equations take the form

$$(3-14) \quad \underline{x}_{n+1} = \underline{F}_n \underline{x}_n + \underline{G}_n \omega_n$$

where (Appendix B)

$$(3-15) \quad \underline{F}_n = \Phi(\Delta) = \begin{bmatrix} \phi_x & 0 & 0 \\ 0 & \phi_y & 0 \\ 0 & 0 & \phi_z \end{bmatrix}$$

with

$$(3-16) \quad \phi_i \quad (i = x, y, z) = \begin{bmatrix} 1 & \Delta & \frac{\Delta}{a_1} + \frac{1}{a_1^2} (e^{-a_1 \Delta} - 1) \\ 0 & 1 & \frac{1}{a_1} (1 - e^{-a_1 \Delta}) \\ 0 & 0 & e^{-a_1 \Delta} \end{bmatrix}$$

and

$$(3-17) \quad \underline{G}_n = \int_0^\Delta \Phi(\tau) \underline{G} d\tau$$

Here, Δ is the time interval between measurements.

The ω_n is a time averaged white noise sequence obtained from the continuous white noise process. That is,

$$(3-18) \quad \omega_i = \frac{1}{\Delta} \int_{n\Delta}^{(n+1)\Delta} u_i(\tau) d\tau \quad (i = 1, 2, 3)$$

The error covariance matrix for the process described by Equation (3-14) is found by evaluating $E(\omega_n \omega_n^T)$ where E is the expectation operator. Assuming $\omega_1, \omega_2, \omega_3$ to be statistically independent, the error covariance matrix defined by Q is

$$(3-19) \quad Q = \begin{bmatrix} E(\omega_1^2) & 0 & 0 \\ 0 & E(\omega_2^2) & 0 \\ 0 & 0 & E(\omega_3^2) \end{bmatrix}$$

From (3-18),

$$E(\omega_i^2) = \frac{1}{\Delta^2} E \int_0^{t_1} \int_0^{t_1} u_i(t) u_i(\tau) dt d\tau$$

$$\begin{aligned}
&= \frac{1}{\Delta^2} \int_0^t \int_0^1 E [u_1(t) u_1(\tau)] dt d\tau \\
&= \frac{1}{\Delta^2} \int_0^t \int_0^1 \delta(t-\tau) dt d\tau = \frac{1}{\Delta}
\end{aligned}$$

where the integration is over the time interval Δ .

Thus,

$$(3-20) \quad Q = \frac{1}{\Delta} I$$

where I is a 3×3 identity matrix.

The autocorrelation function for this process has a very simple form. If $R_1(\tau)$ with $i = x, y, z$ is the autocorrelation function for the target acceleration along the i th coordinate, then (Appendix C)

$$(3-21) \quad R_1(\tau) = \frac{b_1^2}{2a_1} e^{-a_1 |\tau|}$$

Thus, a_1 is related inversely to the correlation time as stated earlier. Furthermore, by setting $\tau = 0$, $R_1(\tau)$ becomes the variance of the acceleration along the i th direction. That is,

$$(3-22) \quad \sigma_1^2 = \frac{b_1^2}{2a_1}$$

Thus, if one has some knowledge of the aircraft RMS acceleration as well as the correlation time for the acceleration process, one can determine the intensity b_1 of the white noise process from the above equation.

So far, no justification has been given for the particular target state model assumed here. The true target acceleration may indeed be a second or higher order Markov process. Unfortunately, the variety of target models is too numerous to make a thorough evaluation of each model practical. Nevertheless, reasonable alternatives to the acceleration process described here are considered in subsequent sections of this report with the conclusion that modeling acceleration randomness as a first order Markov process is indeed adequate.

MODEL OF THE OBSERVATIONS

It is assumed that the sensor provides measurements of R , θ , and φ as defined in Figure 1. Furthermore, the measurements are assumed to be statistically independent quantities whose noise components have zero mean and known values of standard deviation. The observations are taken to be the three components of position measurements in cartesian coordinates with additive white noise. Thus,

$$(3-23) \quad z_1 = x_1 + v_1$$

$$(3-24) \quad z_2 = x_4 + v_2$$

$$(3-25) \quad z_3 = x_7 + v_3$$

The covariance matrix \underline{R} of the observation noise is non-diagonal due to the coupling of the noise terms along x , y , and z through the measurements R , θ , and φ . Indeed, \underline{R} is state dependent because of the dependence of the observation errors v_1 , v_2 , and v_3 upon the target state.

In matrix form, the observation equations are written as

$$(3-26) \quad \underline{z} = \underline{H}\underline{x} + \underline{D}\underline{v}$$

where, by inspection,

$$\underline{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \quad \underline{D} = \underline{I}, \quad \underline{v} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

with \underline{I} a 3 x 3 identity matrix.

By definition,

$$(3-27) \quad \underline{R} = E[\underline{v} \underline{v}^T] = \begin{bmatrix} \sigma_{v_1}^2 & E(v_1 v_2) & E(v_1 v_3) \\ E(v_1 v_2) & \sigma_{v_2}^2 & E(v_2 v_3) \\ E(v_1 v_3) & E(v_2 v_3) & \sigma_{v_3}^2 \end{bmatrix}$$

Given the variances of the errors in R , θ , and φ , namely σ_R^2 , σ_θ^2 , and σ_φ^2 , one obtains, in a straightforward manner, the elements of \underline{R} explicitly. From Appendix D,

$$(3-28) \quad R_{11} = \sigma_{v_1}^2 = R^2 [\sigma_\theta^2 \cos^2 \theta \cos^2 \varphi + \sigma_\theta^2 \sigma_\varphi^2 \cos^2 \theta \sin^2 \varphi + \sigma_\varphi^2 \sin^2 \theta \sin^2 \varphi] + \sigma_R^2 \sin^2 \theta \cos^2 \varphi$$

$$(3-29) \quad R_{12} = R_{21} = E(v_1 v_2) = R^2 \sin^2 \varphi \cos^2 \varphi [\sigma_\theta^2 \cos^2 \theta - \sigma_\varphi^2 \sin^2 \theta] + \sigma_R^2 \sin^2 \theta \sin \varphi \cos \varphi$$

$$(3-30) \quad R_{13} = R_{31} = E(v_1 v_3) = (\sigma_R^2 - R^2 \sigma_\theta^2) \cos \theta \sin \theta \cos \varphi$$

$$(3-31) \quad R_{23} = R_{32} = E(v_2 v_3) = (\sigma_R^2 - R^2 \sigma_\theta^2) \cos \theta \sin \theta \sin \varphi$$

$$(3-32) \quad R_{22} = \sigma_{v2}^2 = R^2 [\sigma_\varphi^2 \sin^2 \theta \cos^2 \varphi + \sigma_\theta^2 \sigma_\varphi^2 \cos^2 \varphi \cos^2 \theta + \sigma_\theta^2 \cos^2 \theta \sin^2 \varphi] \\ + \sigma_R^2 \sin^2 \theta \sin^2 \varphi$$

$$(3-33) \quad R_{33} = \sigma_{v3}^2 = R^2 \sigma_\theta^2 \sin^2 \theta + \sigma_R^2 \cos^2 \theta$$

In implementing the Kalman filter, one may use either current state estimates to determine R , θ , and φ for use in the \underline{R} matrix or raw sensor values. Because these two alternatives give virtually the same numerical values for the matrix elements for \underline{R} , the simpler alternative of using sensor values for R , θ , and φ will be used to compute \underline{R} .

It will be shown in Section 14 that, for the proposed filter design, the elements of \underline{R} can be replaced by numerical constants with little loss in filter performance. Indeed, unless this is done, the gains never reach steady state values. By using steady state gains, one can investigate the performance of an exceedingly simply steady state filter design.

SECTION 4

INITIALIZATION OF THE FILTER

In all filter simulations, a standard procedure described in this section will be used to initialize the state vector \underline{x} as well as the state estimation covariance matrix \underline{P} .

The procedure for initializing \underline{x} is straightforward. Four position measurements are made, where the last measurement is used as the initial value of target position and the first and fourth measurements are used to compute the velocity components. That is, if $z_i(0)$ and $z_i(3\Delta)$ represent the i th component of measured position at times $t = 0$ and $t = 3\Delta$, where Δ is the time between observations, then the j th component of initial velocity state is

$$(4-1) \quad x_j(0) = \frac{z_i(3\Delta) - z_i(0)}{3\Delta}$$

where $j = 2, 5$ or 8 depending on whether i is x, y , or z . The use of four observations is rather arbitrary.

The initial acceleration estimate is set to zero. Position measurements are not used to estimate target acceleration since this procedure yields exceedingly noisy results. Indeed, it is not impossible to be in error by several hundred percent using raw sensor measurements to estimate acceleration. It is therefore, best to initialize the acceleration to its expected value, namely zero.

The elements of the state covariance matrix, \underline{P}_0 , defined by $E[(\hat{\underline{x}}_0 - \underline{x}_0)(\hat{\underline{x}}_0 - \underline{x}_0)^T]$, where $\hat{\underline{x}}_0$ is the initial state estimate, are easily computed.

Since the initial position estimate is a raw sensor position measurement, we have

$$(4-2) \quad P_{11} = E[(\hat{x}_1(0) - x_1(0))^2] = E(z_1 - x_1)^2 = E(v_1^2) = R_{11}$$

$$(4-3) \quad P_{44} = E[(\hat{x}_4(0) - x_4(0))^2] = E(z_2 - x_4)^2 = E(v_2^2) = R_{22}$$

$$(4-4) \quad P_{77} = E[(\hat{x}_7(0) - x_7(0))^2] = E(z_3 - x_7)^2 = E(v_3^2) = R_{33}$$

$$(4-5) \quad P_{14} = P_{41} = R_{12}$$

$$(4-6) \quad P_{17} = P_{71} = R_{13}$$

$$(4-7) \quad P_{47} = P_{74} = R_{23}$$

where the elements of \underline{R} are given in the previous section. The remaining elements of \underline{P}_0 , with the exception of the elements relating to the target acceleration, are computed in Appendix E with the following results:

$$(4-8) \quad P_{12} = P_{21} = \frac{1}{3\Delta} R_{11}$$

$$(4-9) \quad P_{15} = P_{51} = \frac{1}{3\Delta} R_{12}$$

$$(4-10) \quad P_{18} = P_{81} = \frac{1}{3\Delta} R_{13}$$

$$(4-11) \quad P_{41} = P_{14} = \frac{1}{3\Delta} R_{12}$$

$$(4-12) \quad P_{45} = P_{54} = \frac{1}{3\Delta} R_{22}$$

$$(4-13) \quad P_{48} = P_{84} = \frac{1}{3\Delta} R_{23}$$

$$(4-14) \quad P_{72} = P_{27} = \frac{1}{3\Delta} R_{13}$$

$$(4-15) \quad P_{75} = P_{57} = \frac{1}{3\Delta} R_{23}$$

$$(4-16) \quad P_{78} = P_{87} = \frac{1}{3\Delta} R_{33}$$

The elements of \underline{R} are evaluated at the end of the time interval 3Δ (i.e. - after the fourth sensor position measurement). The velocity variance and velocity cross terms are

$$(4-17) \quad P_{22} = \frac{R_{11}(3\Delta) + R_{11}(0)}{(3\Delta)^2}$$

$$(4-18) \quad P_{55} = \frac{R_{22}(3\Delta) + R_{22}(0)}{(3\Delta)^2}$$

$$(4-19) \quad P_{88} = \frac{R_{33}(3\Delta) + R_{33}(0)}{(3\Delta)^2}$$

$$(4-20) \quad P_{25} = P_{52} = \frac{R_{12}(3\Delta) + R_{12}(0)}{(3\Delta)^2}$$

$$(4-21) \quad P_{28} = P_{82} = \frac{R_{13}(3\Delta) + R_{13}(0)}{(3\Delta)^2}$$

$$(4-22) \quad P_{58} = P_{85} = \frac{R_{23}(3\Delta) + R_{23}(0)}{(3\Delta)^2}$$

By admitting total ignorance of target acceleration, the variance of each component of the initial acceleration estimate is set equal to $(3g)^2$ ($g = 9.8\text{m/sec}^2$). Thus,

$$(4-23) \quad P_{33} = P_{66} = P_{99} = (3g)^2$$

with the remaining elements of P_0 set equal to zero.

It is not obvious that this elaborate procedure for initializing the Kalman filter is really necessary. It may be argued that suitable results will be obtained by initializing the filter with some constant pre-determined values. Experience early in the program, however, has shown that some estimates for initial conditions can lead to long settling times and indeed, a bad choice of initial conditions can lead to instability in the Kalman equations resulting in divergence of the elements of \underline{P} . The above procedure eliminates the guesswork associated with initialization of the Kalman filter and is therefore used in all subsequent computer simulations.

SECTION 5

SUBOPTIMAL FILTER DESIGN PHILOSOPHY

The filter described in the preceeding section has nine state variables. Implementation of the filter equations therefore requires matrix inversion at each step of the iteration process. If one can eliminate the need to perform matrix inversion at each iteration step, then one will have reduced significantly the execution time for the solution to the state estimates. Of necessity, however, one must pay the penalty of degrading the accuracy of the state estimates since reduction of the complexity of the filter discussed heretofore generally results in some loss of information, either about the plant or the observation models or both. Nevertheless, if the degradation in filter performance is small, simplification in the filter design may be justified.

A considerable simplification results when one considers the following design modification: Instead of using a single nine state filter in which the three cartesian coordinates are coupled via the polar measurements R, θ, ϕ , design three separate and independent filters, one for each cartesian coordinate. In such a design, information about the coupling is lost, but the decrease in complexity is significant. Thus, the nine state filter is replaced by three independent three state filters. The need to perform matrix inversion is then replaced by inversion of a single number. Henceforth, the filters described will all have this simplifying feature. The only exception is the discussion in Section 21 in which the performance of the nine state and the three state filters is compared.

SECTION 6

THE FIVE STATE FILTER

MODEL OF THE PLANT

Here, we consider a filter in which the acceleration process is modeled as the sum of two terms: a deterministic term and a stochastic term. The stochastic portion of the acceleration is modeled as a first order Markov process. Thus, for each of the three cartesian coordinates, we have the following model of the target motion:

$$(6-1) \quad \dot{x}_1 = x_2$$

$$(6-2) \quad \dot{x}_2 = x_3 + x_4$$

$$(6-3) \quad \dot{x}_3 = -ax_3 + b\omega$$

$$(6-4) \quad \dot{x}_4 = x_5$$

$$(6-5) \quad \dot{x}_5 = 0$$

where x_1 is the target position along one of the three cartesian axes. This model assumes that the rate of change of the deterministic portion of the acceleration is constant.

In matrix form,

$$(6-6) \quad \dot{\underline{x}} = \underline{F}\underline{x} + \underline{G}\omega$$

where, by inspection

$$\underline{F} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & -a & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\underline{G} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \quad \underline{\omega} = \omega$$

Discretizing the plant equation leads to

$$(6-7) \quad x_{n+1} = F_n x_n + G_n \omega_n$$

where (Appendix E),

$$(6-8) \quad \underline{F}_n = \Phi(\Delta) = \begin{bmatrix} 1 & \Delta & \frac{1}{a} \left[\Delta - \frac{1}{a} (1 - e^{-a\Delta}) \right] & \frac{1}{2} \Delta^2 & \frac{1}{6} \Delta^3 \\ 0 & 1 & \frac{1}{a} (1 - e^{-a\Delta}) & \Delta & \frac{1}{2} \Delta^2 \\ 0 & 0 & e^{-a\Delta} & 0 & 0 \\ 0 & 0 & 0 & 1 & \Delta \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$(6-9) \quad \underline{G}_n = \int_0^{\Delta} \Phi(\tau) \underline{G} d\tau = \begin{bmatrix} \frac{\Delta}{a} \left(\frac{\Delta}{2} - \frac{1}{a} \right) + \frac{1}{3a} (1 - e^{-a\Delta}) \\ \frac{1}{a^2} (a\Delta - (1 - e^{-a\Delta})) \\ \frac{1}{a} (1 - e^{-a\Delta}) \\ 0 \\ 0 \end{bmatrix} b$$

and

$$(6-10) \quad \omega_n = \frac{1}{\Delta} \int_{n\Delta}^{(n+1)\Delta} \omega(t) dt$$

The correlation function for the stochastic portion of the acceleration is the same as for the nine state filter. Thus,

$$(6-11) \quad R(\tau) = \frac{b^2}{2a} e^{-a|\tau|}$$

From the equations for the state variables x_4 and x_5 ,

$$(6-12) \quad x_5 = \alpha$$

$$(6-13) \quad x_4 = \beta + \alpha t$$

At $t = 0$, $x_4 = \beta$ and $x_5 = \alpha$. These are set equal to zero for \hat{x}_0 .

MODEL OF THE SENSOR

With obvious minor modifications in \underline{H} , the observation model is as described in Section 3. Unless otherwise stated, the sensor is henceforth characterized by a range variance σ_R^2 of $100m^2$ and angular error variances σ_θ^2 , σ_ϕ^2 each of $1m^2$. It will be shown in Section 14, however, that the elements of \underline{R} can be replaced by constant elements with no degradation in filter performance.

SECTION 7

MODIFIED FIVE STATE FILTER - THE FOUR STATE FILTER

An obvious modification of the five state filter results when Equation (6-4) is replaced by $x_4 = 0$. That is, the deterministic portion of the acceleration is modeled as being constant. This results in a reduction in the number of state variables from five to four. The four state target model is then

$$(7-1) \quad \dot{x}_1 = x_2$$

$$(7-2) \quad \dot{x}_2 = x_3 + x_4$$

$$(7-3) \quad \dot{x}_3 = -ax_3 + b_0$$

$$(7-4) \quad \dot{x}_4 = 0$$

The F and G matrices are modified in an obvious way, with corresponding modifications in F_n and G_n . Their explicit form will, therefore, not be given here.

SECTION 8

RESULTS

The fourth and fifth order filters for the y component of target motion were optimized for flight passes 1 and 9 (Tables 2 and 3). Optimization proceeded in the following way: From Equation (6-11), $R(0) = \sigma^2$, where σ^2 is the variance of target acceleration (in this case, along the y coordinate). Then,

$$(8-1) \quad \sigma^2 = \frac{b^2}{2a}$$

Given σ^2 and a, one can then solve for the parameter b^2 . For a given value of a, it was determined that, in general, an acceleration variance of $1000 \text{ (m/sec}^2\text{)}^2$ results in best filter performance. This value of σ^2 is used throughout the report. For each value of a, the RMS of the target position, (in this case the y component), velocity, and acceleration errors ϵ_y , $\epsilon_{\dot{y}}$, $\epsilon_{\ddot{y}}$ is computed over the entire run. The filter is said to be optimized when a value of a is chosen which results in the smallest RMS values for ϵ_y , $\epsilon_{\dot{y}}$, $\epsilon_{\ddot{y}}$. The value of a which optimizes filter performance is henceforth called the optimal value of a.

TABLE 2

COMPARISON BETWEEN THE FOURTH AND FIFTH ORDER AND
FILTERS ALONG THE y COORDINATE FOR PASS NO. 1

<u>a</u>		<u>RMS(y-ŷ)</u>		<u>RMS(ẏ-ŷ̇)</u>		<u>RMS(ÿ-ŷ̈)</u>	
Orders		Orders		Orders		Orders	
4th	5th	4th	5th	4th	5th	4th	5th
.001	.001	9.3	8.9	14.2	14.4	11.1	11.7
.01	.01	5.3	5.3	9.3	9.5	9.5	9.9
.05*	.05*	4.9	4.9	8.3	8.5	8.9	9.3
.08	.08	5.0	5.0	8.5	8.6	8.9	9.2
.1	.1	5.0	5.0	8.6	8.7	9.0	9.2
.5	.5	5.4	5.4	10.2	10.2	10.8	10.9
1.2	1.2	5.5	5.5	10.8	10.9	12.4	12.5

TABLE 3
COMPARISON BETWEEN THE FOURTH AND FIFTH ORDER AND
FILTERS ALONG THE y COORDINATE FOR PASS NO. 9

<u>a</u>	<u>a</u>	<u>RMS($y-\hat{y}$)</u>		<u>RMS($\dot{y}-\dot{\hat{y}}$)</u>		<u>RMS($\ddot{y}-\ddot{\hat{y}}$)</u>	
Orders		Orders		Orders		Orders	
4th	5th	4th	5th	4th	5th	4th	5th
.001	.001	11.6	11.3	17.5	17.6	13.5	13.9
.01	.01	6.6	6.5	11.2	11.3	11.4	11.7
.07*	.07*	5.8	5.8	9.9	10.1	10.6	10.9
.12	.12	5.8	5.8	10.1	10.3	10.7	10.9
1.0	1.0	6.0	6.0	11.3	11.4	12.5	12.3
1.2	1.2	5.5	5.5	10.8	10.9	12.5	12.5

As seen in Tables 2 and 3, there is very little difference in performance between the 4th and 5th order filters. Furthermore, the optimal values of a (those with an asterisk) for both filters are the same. More extensive runs for these filters were not made in view of the results obtained for the filter proposed for use in air defense (see Section 15). That is, the 4th and 5th order filters, which involve more state variables than the proposed filter, do not offer improved performance.

SECTION 9

PROCESS IDENTIFICATION

As mentioned previously, a value for the parameter "a" appearing in the equation for \dot{x}_3 (Equations (6-3), (7-3)) can be obtained from theoretical considerations. In this report, two basic acceleration models for the stochastic portion of the acceleration are investigated - a first order Markov model and a second order Markov model. Consider the first order Markov model

$$(9-1) \quad \dot{x} = -ax + b\omega.$$

In discrete form,

$$(9-2) \quad x_{n+1} = x_n - \Delta ax_n + \Delta b\omega_n$$

where Δ is a small time increment and the subscripted integer n means that the variable to which it is affixed is evaluated at the time $n\Delta$. Observe that x_n is independent of ω_n . Multiplying Equation (9-2) by x_n and taking the expected value of both sides, one obtains

$$(9-3) \quad E(x_n x_{n+1}) = (1 - \Delta a) E(x_n^2)$$

where the assumption that the mean of the white noise sequence ω_n is zero was used. Observe that the expectation operator acts over an ensemble. Now assume that the process is stationary and ergodic. Then, the ensemble statistics are the same as the statistics over time. Specifically,

$$(9-4) \quad E(x_n x_{n+k}) = \phi(k)$$

for all n where $\phi(k)$ is the correlation function. Thus,

$$(9-5) \quad \phi(\Delta) = (1 - \Delta a) \phi(0)$$

or

$$(9-6) \quad a = \frac{1 - \phi(\Delta)}{\Delta}$$

where $\phi(0)$ is normalized to unity.

The procedure for obtaining expressions for the appropriate variables in the second order Markov model is similar to the one above. The same assumptions about ergodicity and the mean of the white noise sequence are made.

The model for the second order Markov process is

$$(9-7) \quad \dot{x}_3 = x_4 + \omega$$

$$(9-8) \quad \dot{x}_4 = -\alpha^2 x_3 - 2\beta x_4 + (\alpha - 2\beta)\omega.$$

(The reason for this particular form of the model is given in Section 12).
Discretize each equation to obtain

$$(9-9) \quad x_3(n+1) = x_3(n) + \Delta x_4(n) + \Delta \omega(n)$$

$$(9-10) \quad x_4(n+1) = x_4(n) + \alpha^2 \Delta x_3(n) - 2\beta \Delta x_4(n) + (\alpha - 2\beta) \Delta \omega(n)$$

Now solve for x_3 and drop the subscript 3. That is, define $x_n = x_3(n)$. Then,

$$(9-11) \quad x_{n+2} + 2(\beta \Delta - 1)x_{n+1} + (1 + \alpha^2 \Delta^2 - 2\beta \Delta)x_n - \Delta \omega_{n+1} + (\Delta - \Delta^2 \alpha)\omega_n = 0$$

Multiplying through by x_n and taking the expected value of both sides, one obtains

$$(9-12) \quad \phi(2\Delta) + 2(\beta \Delta - 1)\phi(\Delta) + (1 + \alpha^2 \Delta^2 - 2\beta \Delta) = 0$$

A second equation involving α and β can be obtained by multiplying Equation (9-11) by x_{n-1} then taking the expected value of the resulting expression. The result is

$$(9-13) \quad \phi(3\Delta) + 2(\beta \Delta - 1)\phi(2\Delta) + (1 + \alpha^2 \Delta^2 - 2\beta \Delta)\phi(\Delta) = 0$$

Solution of Equations (9-12) and (9-13) for α and β gives

$$(9-14) \quad \beta = \frac{P_1 + 2}{2\Delta}$$

and

$$(9-15) \quad \alpha^2 = \frac{2\beta \Delta + P_2 - 1}{\Delta^2}$$

where

$$(9-16) \quad P_1 = \frac{\phi(\Delta)\phi(2\Delta) - \phi(3\Delta)}{\phi(2\Delta) - \phi(\Delta)^2}$$

$$(9-17) \quad P_2 = -\phi(2\Delta) - P_1 \phi(\Delta)$$

The correlation function $\phi(k)$ is computed from the definition

$$(9-18) \quad \phi(k) = \frac{1}{\phi(0)N} \sum_{i=1}^{N-k} x_i x_{i+k}$$

where N is the number of available data points.

Tables 5 through 7 list the computed values of "a" using Equation (9-6) for each of the 12 FACT flight passes. With ω_0 defined by $\omega_0^2 = \alpha_0^2 - \beta^2$, Tables 8 through 10 list the computed values of ω_0 and β which parameterize the second order Markov acceleration process. Table 4 lists the number of data points in each flight path as well as the number of data points in each of the three segments for each flight path. Each segment is characterized by a specific portion of the flight path. Segment 1 defines that portion of the flight path during which the aircraft undergoes evasive maneuvers prior to weapon delivery. Segment 2 is characterized by that portion of the flight path during which the pilot is constrained to follow a particular course in order to deliver his ordinance on target. It is taken to start 5 seconds before weapon release. Segment 3 is the remaining portion of the flight path. Mean acceleration values for each of the three portions as well as the standard deviation of the acceleration along the entire run are also provided in Tables 5, 6, and 7. The data points are separated by 1/10 second intervals.

TABLE 4
NUMBER OF DATA POINTS FOR EACH FLIGHT PATH AND
FOR EACH SEGMENT OF A FLIGHT PATH

<u>Flight Pass</u>	<u>N</u>	<u>N</u>	<u>N</u>	<u>N</u>
<u>Number</u>	<u>Total</u>	<u>Seg 1</u>	<u>Seg 2</u>	<u>Seg 3</u>
1	365	142	51	172
2	483	329	51	103
3	383	221	51	111
4	412	261	51	100
5	405	329	51	25
6	345	206	51	88
7	378	220	51	107
8	331	171	51	109
9	496	381	51	64
10	573	443	51	79
11	576	496	51	29
12	589	538	51	0

TABLE 5

THEORETICAL VALUES OF "a" FOR x COMPONENT OF ACCELERATION MODELED AS A FIRST ORDER MARKOV PROCESS

Mean acceleration \bar{x}_3 (m/sec²) as well as standard deviation of acceleration $\sigma_{\bar{x}_3}^2$ (m/sec²)² along x are also listed.

Flight Pass Number	a			\bar{x}_3			$\sigma_{\bar{x}_3}^2$		
	Entire Run	Seg 1	Seg 2	Seg 3	Entire Run	Seg 1	Seg 2	Seg 3	Entire Run
1	.056	.099	.423	.106	- 9.5	-13.6	- 1.0	- 8.5	204
2	.041	.039	.514	.141	- 3.4	- 4.5	-12.4	- 2.8	106
3	.061	.103	.658	.108	- 1.7	0.0	9.8	-10.9	577
4	.070	.046	.202	.272	- 6.3	- 8.2	-23.7	7.7	238
5	.021	.018	.812	1.346	2.1	- 1.8	10.7	30.3	236
6	.038	.067	1.102	.085	- 2.7	-11.9	- 2.9	19.8	295
7	.058	.034	2.310	.096	-19.0	-12.3	- 2.1	-12.9	315
8	.072	.070	2.239	.091	-14.3	-14.1	- 4.0	1.5	266
9	.036	.228	.324	.422	- 4.2	2.2	-18.6	-30.7	407
10	.034	.033	.614	.276	- 3.5	.6	17.1	-33.9	347
11	.021	.024	.255	.797	- 3.2	- 1.9	-12.8	- 8.0	519
12	.017	.017	.348	*	- .4	- .4	.4	*	32

* No data points available for computation.

TABLE 6

THEORETICAL VALUES OF "a" FOR y COMPONENT OF ACCELERATION MODELED AS A FIRST ORDER MARKOV PROCESS

Mean acceleration \bar{y}_3 (m/sec²) as well as standard deviation of acceleration $\sigma_{y_3}^2$ (m/sec²)² along y are also listed.

Flight Pass Number	a			\bar{y}_3			$\sigma_{y_3}^2$		
	Entire Run	Seg 1	Seg 2	Seg 3	Entire Run	Seg 1	Seg 2	Seg 3	Entire Run
1	.034	.110	.813	.043	-3.4	2.9	7.0	-11.7	252
2	.019	.018	.584	.187	1.2	2.8	5.5	-7.6	27
3	.031	.093	.858	.053	1.9	1.9	11.5	-2.4	97
4	.032	.034	.164	.392	-4.5	4.3	.3	-29.8	271
5	.035	.033	.954	1.569	2.6	2.8	7.7	-19.3	53
6	.166	.150	.320	.280	3.6	6.7	12.1	-14.4	33
7	.029	.023	.374	.704	-1.5	6.3	9.7	-20.5	167
8	.043	.044	.470	.045	5.8	5.4	11.5	-4.7	94
9	.074	.035	.196	.473	-4.3	.2	-	-32.4	230
10	.034	.027	.718	.222	-4.7	-.7	-11.5	-14.9	123
11	.052	.074	.334	.724	-2.5	-.3	-16.2	-15.0	83
12	.044	.044	2.419	*	-.1	0.0	-.7	*	1

* No data points available for computation.

TABLE 7

THEORETICAL VALUES OF "a" FOR z COMPONENT OF ACCELERATION MODELED AS A FIRST ORDER MARKOV PROCESS

Mean acceleration $\bar{x}_3(\text{m/sec}^2)$ as well as standard deviation of acceleration $\sigma_{z_3}^2(\text{m/sec}^2)^2$ along z are also listed.

Flight Pass Number	a		a		a		\bar{z}_3		\bar{z}_3		\bar{z}_3		$\sigma_{z_3}^2$	
	Entire Run	Seg 1	Seg 2	Seg 3	Entire Run	Seg 1	Seg 2	Seg 3	Entire Run	Seg 1	Seg 2	Seg 3	Entire Run	Seg 3
1	.008	.128	.310	.180	4.0	-9.1	-3.3	17.0	238					
2	.031	.093	.279	.358	3.9	-3.6	6.8	20.9	95					
3	.024	.121	.420	.407	1.8	-7.4	18.9	12.5	163					
4	.016	.052	.625	.298	3.2	-7.0	21.2	20.5	289					
5	.034	.101	.244	.453	-1.9	-5.3	18.4	6.8	35					
6	.026	.090	.246	.492	6.6	-8.4	26.6	19.5	294					
7	.014	.034	.212	.313	5.0	-6.0	24.7	10.8	181					
8	.023	.110	.518	.238	4.4	-9.0	13.8	14.5	239					
9	.023	.033	.663	.394	2.5	1.4	13.4	0.0	425					
10	.025	.028	1.000	.520	1.8	.3	29.1	-7.0	410					
11	.030	.041	.709	.960	2.6	2.4	8.6	-4.4	217					
12	.041	.041	.163	*	.5	.5	-.3	*	21					

* No data points available for computation.

TABLE 8

THEORETICAL VALUES OF ω_0 AND β FOR x COMPONENT OF ACCELERATION MODELED AS A SECOND ORDER MARKOV PROCESS

<u>Flight Pass</u>	<u>ω_0</u>	<u>ω_0</u>	<u>ω_0</u>	<u>ω_0</u>	<u>β</u>	<u>Seg 1</u>	<u>Seg 2</u>	<u>Seg 3</u>
<u>Number</u>	<u>Entire Run</u>	<u>Seg 1</u>	<u>Seg 2</u>	<u>Seg 3</u>	<u>Entire Run</u>	<u>Seg 1</u>	<u>Seg 2</u>	<u>Seg 3</u>
1	.556	.461	.831	.563	.149	.258	.264	.158
2	.577	.581	.551	.441	.271	.262	.292	-.075
3	.563	.215	1.126	.557	.096	.527	.402	.135
4	.564	.778	.986	*	.133	.147	.141	-4.419
5	.401	.414	1.089	1.251	.129	.126	.712	.804
6	.851	.994	*	.649	.122	.151	-6.678	.075
7	.454	.487	1.836	.577	.074	.083	1.379	.507
8	.795	.795	*	.517	.087	.089	3.613	.139
9	.607	.888	.826	*	.163	.339	.082	-.055
10	.670	.690	.720	.628	.077	.079	.223	.292
11	.625	.625	.938	1.33	.075	.082	.410	.147
12	.565	.565	1.155	**	.079	.079	.687	**

* Computation of ω_0 results in imaginary value.

** No data points available for computation.

TABLE 9

THEORETICAL VALUES OF w_0 AND β FOR y COMPONENT OF ACCELERATION MODELED AS A SECOND ORDER MARKOV PROCESSFlight Pass

Number	Entire Run	Seg 1	Seg 2	Seg 3	Entire Run	Seg 1	Seg 2	Seg 3
1	.389	.842	1.079	.351	.136	.208	.537	.036
2	.482	.499	.733	1.108	.131	.178	.178	.273
3	.756	*	1.845	.859	.128	-.467	.525	.084
4	.432	.688	1.175	.683	.219	.181	.095	.751
5	.441	.441	1.149	1.159	.133	.133	.401	.915
6	.948	1.117	1.372	.619	.299	.291	.182	-.039
7	.521	.658	.560	.969	.087	.081	.472	.125
8	.819	.822	2.124	.612	.158	.157	.682	.151
9	.532	.622	1.332	.374	.083	.253	.190	.072
10.	.592	.635	1.228	.634	.165	.156	.558	.442
11.	.770	.938	1.035	.893	.099	.105	.182	-.311
12.	.706	.708	.359	**	.165	.166	-.065	**

* Computation of w_0 results in imaginary value.

** No data points available for computation.

TABLE 10

THEORETICAL VALUES OF ω_0 AND β FOR z COMPONENT OF ACCELERATION MODELED AS A SECOND ORDER MARKOV PROCESSFlight Pass

<u>Number</u>	<u>Entire Run</u>	<u>Seg 1</u>	<u>Seg 2</u>	<u>Seg 3</u>	<u>Entire Run</u>	<u>Seg 1</u>	<u>Seg 2</u>	<u>Seg 3</u>
1	.297	.344	1.516	.622	.183	.372	.547	.166
2	.389	.653	.951	.763	.200	.081	.165	.249
3	.413	*	.691	1.132	.218	-.833	.649	.090
4	.364	.428	1.286	*	.269	.259	.285	.779
5	.532	.655	1.252	1.292	.150	.120	.235	.898
6	.412	.999	.929	.787	.255	.202	.184	.058
7	.288	.491	.947	*	.090	.119	.197	-11.28
8	.392	.822	1.471	*	.113	.187	.569	.949
9	.549	.559	1.024	2.111	.218	.237	.691	1.144
10	.483	.485	1.121	*	.185	.187	.795	-2.167
11	.658	.686	*	1.073	.134	.111	-2.649	.567
12	.874	.874	1.444	**	.243	.243	.402	**

* Computation of ω_0 results in imaginary value.

** No data points available for computation.

Information about the values of "a" and α and β is useful because these values provide some clue as to where to start the search to obtain their optimal values. Furthermore, the range over which these parameters vary from flight path to flight path gives some indication of the sensitivity of a particular model to changes in flight characteristics. By partitioning each flight path into three segments, one can also examine the sensitivity of a model to changes in aircraft motion within a single flight path. This leads, in a natural way, to considerations of adoptive filters where parameters such as "a" are adjusted during the course to take into account changes in aircraft maneuvers. Finally, one can observe the difference between the theoretical values of model parameters and their optimal values. If pronounced differences occur, this is an indication that the assumptions of ergodicity and stationarity of target acceleration are not valid. Comparison of the optimal values of "a" for the 4th and 5th order filters of the previous section with the theoretical values shows that the two sets agree favorably. However, this is not always the case, especially when one considers variations of the parameters over each of the three segments. This variation will be exhibited when the three state, first order Markov and the four state, second order Markov filters are compared in the next section.

SECTION 10

THE THREE AND FOUR STATE FILTERS

INTRODUCTION

The results of Section 8 indicate that there is little difference between the performance of four and five state filters where the stochastic portion of the acceleration is modeled as a first order Markov process. In this section, a simple three state filter is developed for which the acceleration is modeled as a first order Markov process. In Section 13, it is compared with a four state filter for which the acceleration is modeled as a second order Markov process. It will be shown that;

a. There is little difference in performance between the three state filters and the four and five state filters of Sections 6 and 7.

b. The difference in performance between the three state filter and the four state, second order Markov filter, is such that the use of the four state filter over the three state filter is unwarranted.

THE THREE STATE FILTER

The three state filter is based on the following model of the target motion along each of the three coordinates:

$$(10-1) \quad \dot{x}_1 = x_2$$

$$(10-2) \quad \dot{x}_2 = x_3$$

$$(10-3) \quad \dot{x}_3 = -ax_3 + bu$$

where x_1 is the target position along one of the three orthogonal axes. The optimal value of "a" may be different for the x and z axes. The target state equations corresponding to motion along x and y should be parameterized by the same numerical value of "a" for reasons discussed in Section 3.

Discretization of these equations leads to the matrix equation

$$(10-4) \quad \underline{x}_{n+1} = \underline{F}_n \underline{x}_n + \underline{G}_n \omega_n$$

where, by analogy with the nine state filter described in Section 3, (see also Appendix B).

$$\underline{F}_n = (\Delta) = \begin{bmatrix} 1 & \Delta & \frac{\Delta}{a} + \frac{1}{a^2} (e^{-a\Delta} - 1) \\ 0 & 1 & \frac{1}{a} (1 - e^{-a\Delta}) \\ 0 & 0 & e^{-a\Delta} \end{bmatrix}$$

and

$$\underline{G} = \int_0^{\Delta} \phi(\tau) \underline{G} d\tau = \begin{bmatrix} \frac{1}{a} \left[\frac{\Delta^2}{2} - \frac{\Delta}{a} - \frac{1}{a^2} (e^{-a\Delta} - 1) \right] \\ \frac{1}{a^2} (e^{-a\Delta} - 1) + \Delta/a \\ - \frac{1}{a} (e^{-a\Delta} - 1) \end{bmatrix}$$

with

$$\underline{G} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

The quantity ω_n is a time averaged white noise sequence (Equation 3-18), and the error covariance matrix $Q = E(\omega_n^2) = b^2/\Delta$.

Before proceeding to a description of the four state filter, it is useful to motivate the design of this filter by examining the way in which the acceleration data is correlated.

SECTION 11

AUTOCORRELATION OF THE FACT ACCELERATION DATA

In the filters described thus far, the acceleration correlation function has the form of a decaying exponential. Although intuition suggests this form to be appropriate, an appreciation for the form of the correlation function for real flight paths can be obtained from the FACT data.

Figure 2 through 4 are plots of the acceleration correlation function for flight paths 1, 9, and 12 along each of the three coordinates. These plots were obtained using the normalized correlation function

$$(11-1) \quad \phi(k) = \frac{1}{N\phi(0)} \sum_{n=1}^{N-k} [x_3(n) - \bar{x}_3(n)][x_3(n+k) - \bar{x}_3(n+k)]$$

Strictly speaking, one should compute the correlation function over an ensemble of flight passes. Since 12 hardly constitutes an adequate sample size to analyze ensemble statistics, one uses Equation (11-1) with the view that general trends in the shape of the curves should provide a clue to the form of the acceleration correlation function. Examination of Figures 2 through 4 indicates that, in general, the correlation function exhibits an oscillatory behavior. This type of behavior is more pronounced for the x and y coordinates than the z coordinate.

In view of these considerations, a four state second order Markov filter will be constructed in such a way as to assure the correlation function for the acceleration process to be an exponentially damped sinusoid. Such an option, it must be noted, does not exist for the three state, first order Markov filter.

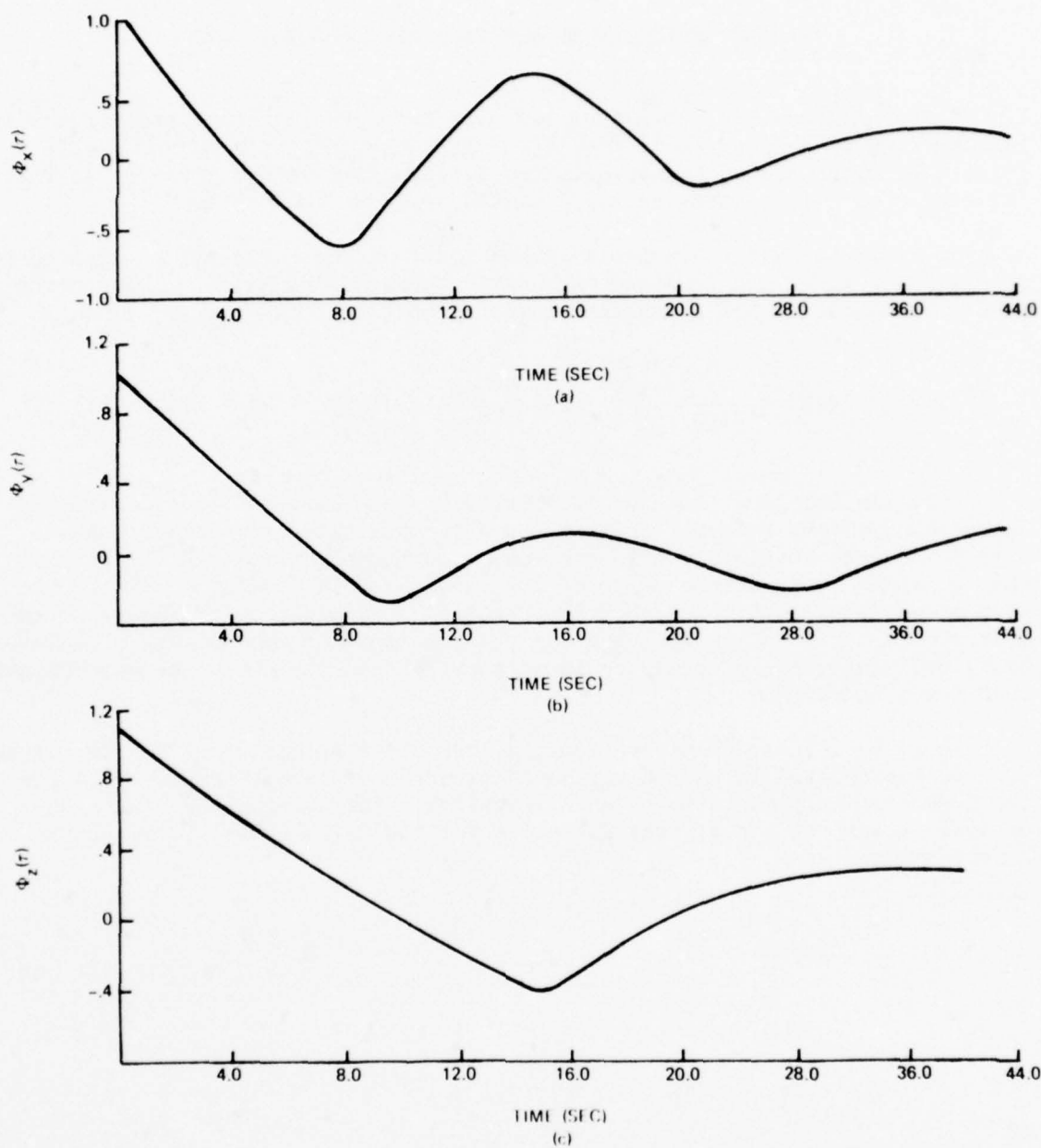


Figure 2. Acceleration correlation functions for flight pass 1

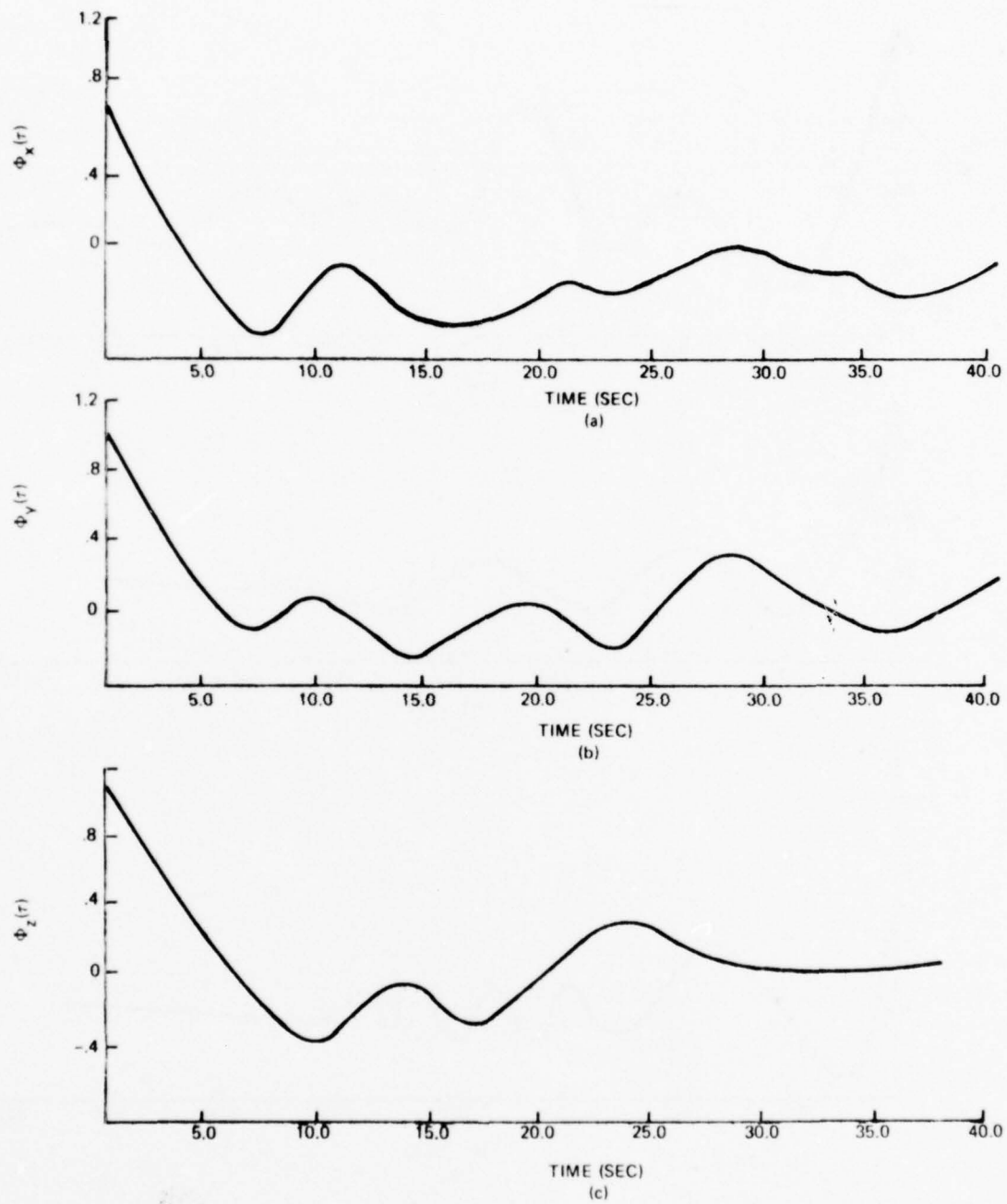


Figure 3. Acceleration correlation functions for flight pass 9

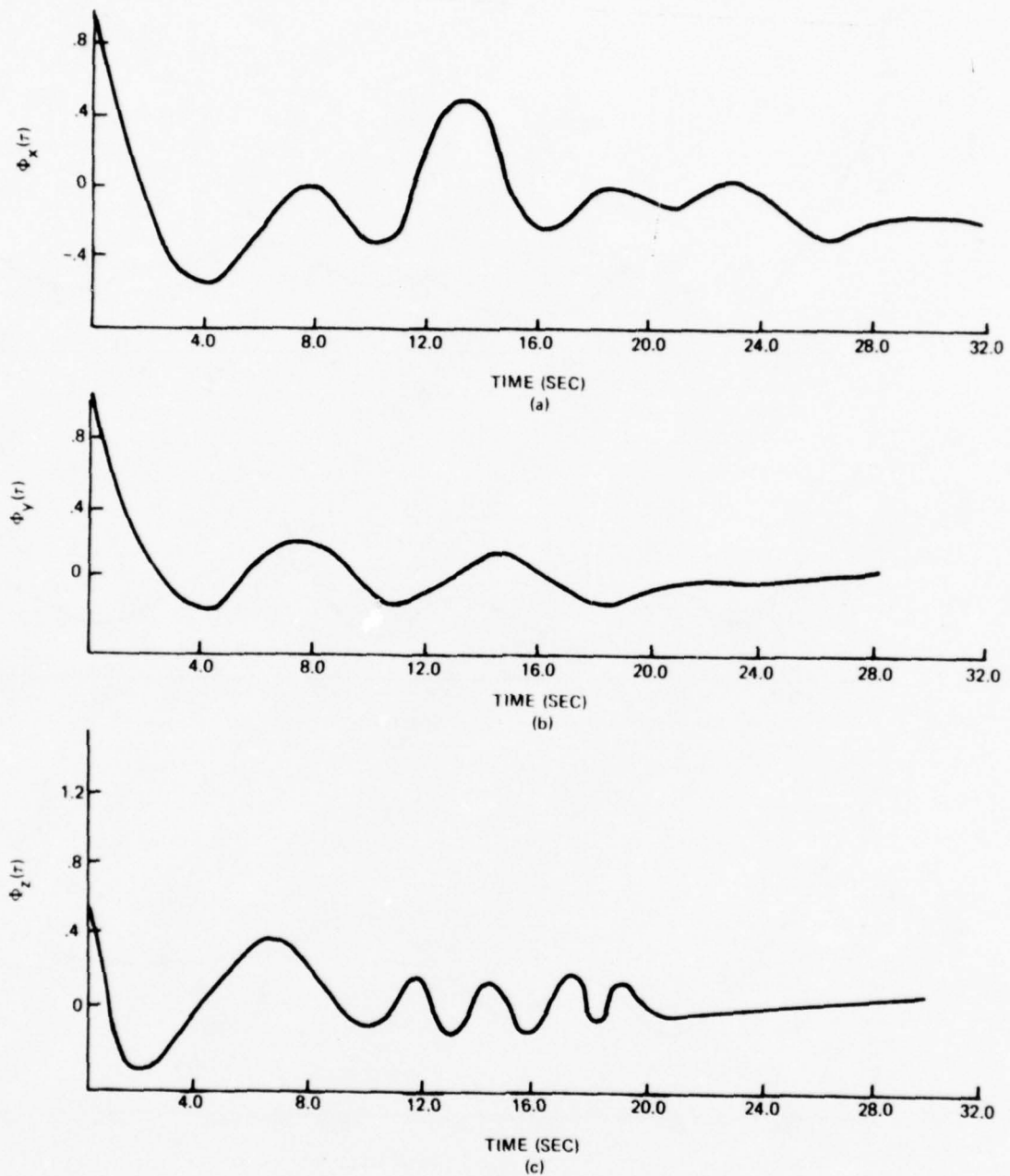


Figure 4. Acceleration correlation functions for flight pass 12

SECTION 12

THE FOUR STATE FILTER

The four state filter is based on a plant model in which the acceleration is described by a second order Markov process. Specifically, the state equations are:

$$(12-1) \quad \dot{x}_1 = x_2$$

$$(12-2) \quad \dot{x}_2 = x_3$$

$$(12-3) \quad \dot{x}_3 = x_4 + bu$$

$$(12-4) \quad \dot{x}_4 = -\alpha^2 x_3 + 2\beta x_4 + (\alpha - 2\beta)b_u$$

where

$$b = \sqrt{2\beta\sigma^2}$$

$$\alpha^2 = \beta^2 + \omega_0^2$$

With this particular form of the state equations, the desired autocorrelation function is obtained (see Appendix C).

$$(12-5) \quad R(\tau) = \frac{b^2}{2\beta} e^{-\beta\tau} \cos \omega_0 \tau$$

The σ^2 appearing in the expression for b is just $R(0)$.

The state transition matrix F_n can be obtained by solving the homogeneous part of the state equations as was done for the previous filters. However, this procedure is algebraically laborious, so the state transition matrix is obtained in the following alternative manner. Start with the homogeneous state equations

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = x_3$$

$$\dot{x}_3 = x_4$$

$$\dot{x}_4 = -\alpha^2 x_3 - 2\beta x_4$$

In matrix form,

$$(12-6) \quad \dot{\underline{x}} = \underline{F} \underline{x}$$

where, by inspection

$$F = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\alpha^2 & -2\beta \end{bmatrix}$$

The solution to Equation (12-6) is

$$(12-7) \quad x(t) = e^{Ft} x(0) = (I + Ft + \frac{F^2 t^2}{2!} + \frac{F^3}{3!} t^3 + \dots) x(0)$$

But the term in parenthesis is just the desired state transition matrix $\phi(t)$.

$$(12-8) \quad \phi(t) = (I + Ft + \frac{F t}{2!} + \dots)$$

In implementing this equation, only the first five terms of the expression are used. Notice that what is actually needed is $F_n = \phi(\Delta)$ where $\Delta = 1/10$ sec. Since α and β are generally small the series converges rapidly.

The non-homogeneous part of the discrete plant model is the term $G_n \omega_n$, where, as shown in Appendix B

$$(12-9) \quad G_n = \int_0^{\Delta} \phi(t) G dt$$

and, by inspection of Equations (12-1) through (12-4),

$$(12-10) \quad G = \begin{bmatrix} 0 \\ 0 \\ b \\ b(\alpha - 2\beta) \end{bmatrix}$$

SECTION 13

RESULTS

The three and four state y coordinate filters were optimized for flight passes 1, 9, and 12. In addition, the filters were optimized for each of three segments of a pass (Section 9). The optimal parameter values for each of the filters as well as filter performance measured by RMS values of errors ϵ_y , $\epsilon\dot{y}$, $\epsilon\ddot{y}$ in position, velocity, and acceleration estimates respectively are tabulated in Table 11. Tables 12 through 17 exhibit filter performance for various parameter values. The optimal values of filter parameters together with filter performance characteristics for these values in terms of RMS state estimate errors are denoted by an asterisk in each table.

From Table 11, it is evident that there is significant variation in the values of the optimal parameters not only among different flight passes, but also among segments of a single flight pass. Variations of optimal parameters for different segments suggest that an adoptive filter may be designed in such a way that the filter parameters are adjusted in accordance with changes in aircraft maneuvers. Parameter variations among different flight paths suggest that there may be some difficulty in choosing a single parameter (two parameters, ω_0 and β , in the case of the four state filter) for which to operate the filter for all flight passes. However, the sensitivity of filter performance to changes in parameter values, as exhibited in Tables 12 through 17, is rather mild so that a decision upon a single parameter value should not be too difficult to make.

Comparison of the RMS values listed in Table 2 of Section 8 of the four state filter and the RMS values listed in Table 11 of the three and four state (second order Markov) filters indicate that there is little difference in performance between these filters. Furthermore, a cursory examination of Table 11 indicates that there are negligible differences in performance between the three and four state (second order Markov) filters. On the basis of this comparison, we tentatively conclude that the three state filter is the cheapest way to go (in terms of computer software) with little or no degradation in performance over the other filter designs.

TABLE 11
COMPARISON OF THREE STATE AND FOUR STATE FILTERS

THREE STATE			FOUR STATE		
Flight Pass 1	$\frac{a}{\epsilon y}$	RMS $\frac{\epsilon \dot{y}}{\epsilon \ddot{y}}$	$\frac{w_o}{\beta}$	RMS $\frac{\epsilon \dot{y}}{\epsilon \ddot{y}}$	RMS $\frac{\epsilon \dot{y}}{\epsilon \ddot{y}}$
Entire	.045	4.9	.175	4.6	8.9
Seg 1	.01	3.5	.2	3.9	7.0
Seg 2	.0005	2.6	.1	2.8	4.4
Seg 3	.04	4.4	.2	4.5	8.4
Flight Pass 9					
Entire	.07	5.8	.005	5.9	10.7
Seg 1	.035	4.7	.005	4.8	7.7
Seg 2	7.0	3.0	1.4	3.0	10.0
Seg 3	.1	4.1	.1	3.9	14.4
Flight Pass 12					
Entire	.001	4.6	.7	4.0	2.2
Seg 2	10	2.4	8.0	2.4	5.1

TABLE 12

OPTIMIZATION OF THREE STATE FILTER (y COMPONENT) FOR FLIGHT PASS 1

\underline{a}	$\underline{\text{RMS}} (\epsilon_y)$	$\underline{\text{RMS}} (\epsilon_{\dot{y}})$	$\underline{\text{RMS}} (\epsilon_{\ddot{y}})$
.02	5.0	8.5	9.2
.04	4.9	8.3	8.9
.045*	4.9	8.3	8.9
.05	4.9	8.3	8.9
.10	5.0	8.6	9.0

TABLE 13

OPTIMIZATION OF FOUR STATE FILTER (y COMPONENT) FOR FLIGHT PASS 1

$\underline{\omega_0}$	$\underline{\beta}$	$\underline{\text{RMS}} (\epsilon_y)$	$\underline{\text{RMS}} (\epsilon_{\dot{y}})$	$\underline{\text{RMS}} (\epsilon_{\ddot{y}})$
.15	.025	4.8	8.3	9.0
.175*	.025	4.8	8.2	8.9
.2	.0075	4.9	8.9	9.2
.2	.01	4.9	8.6	9.2
.2	.025	4.8	8.2	8.9
.2	.05	4.9	8.3	8.9
.2	.10	5.0	8.7	4.2
.2	.12	5.1	8.8	9.3
.2	.14	5.1	9.0	9.4
.2	.16	5.1	9.1	9.6
.2	.18	5.2	9.2	9.6
.3	.10	5.1	8.9	9.5
.3	.12	5.1	9.0	9.6

TABLE 13 (Cont)

ω_0	β	RMS (ϵ_y)	RMS ($\epsilon_{\dot{y}}$)	RMS ($\epsilon_{\ddot{y}}$)
.3	.14	5.1	9.2	9.7
.3	.16	5.2	9.3	9.9
.3	.18	5.2	9.2	10.0
.4	.10	5.2	9.2	10.2
.4	.12	5.2	9.3	10.3
.4	.14	5.2	9.4	10.4
.4	.16	5.2	9.6	10.5
.4	.18	5.3	9.6	10.6
.5	.10	5.3	9.6	11.3
.5	.12	5.3	9.7	11.3
.5	.14	5.3	9.8	11.4
.5	.16	5.3	9.9	11.4
.6	.10	5.4	10.1	12.7
.6	.12	5.4	10.1	12.6
.6	.14	5.4	10.2	12.6
.6	.16	5.4	10.3	12.6
.6	.18	5.4	10.3	12.6

TABLE 14

OPTIMIZATION OF THREE STATE FILTER (y COMPONENT) FOR FLIGHT PASS 9

a	RMS (ϵ_y)	RMS ($\epsilon_{\dot{y}}$)	RMS ($\epsilon_{\ddot{y}}$)
.005	7.4	12.6	12.0
.01	6.6	11.2	11.4
.04	5.9	9.9	10.7

TABLE 14 (Cont)

\underline{a}	$\underline{\text{RMS } (\epsilon_y)}$	$\underline{\text{RMS } (\epsilon_{\dot{y}})}$	$\underline{\text{RMS } (\epsilon_{\ddot{y}})}$
.07*	5.8	9.9	10.6
.1	5.8	10.1	10.7

TABLE 15

OPTIMIZATION OF FOUR STATE FILTER (y COMPONENT) FOR FLIGHT PASS 9

$\underline{\omega_o}$	$\underline{\beta}$	$\underline{\text{RMS } (\epsilon_y)}$	$\underline{\text{RMS } (\epsilon_{\dot{y}})}$	$\underline{\text{RMS } (\epsilon_{\ddot{y}})}$
.005	.2	5.8	10.5	11.0
.0001	.16	5.8	10.3	10.9
.0005	.16	5.8	10.3	10.9
.001	.16	5.8	10.3	10.9
.005	.16	5.8	10.3	10.9
.03	.16	5.8	10.3	10.9
.06	.16	5.8	10.3	10.9
.09	.16	5.8	10.4	10.9
.10	.16	5.8	10.4	11.0
.3	.16	5.8	10.6	11.2
.6	.16	5.9	11.1	12.4
.9	.16	6.1	11.9	15.3
.005	.10	5.8	10.1	10.7
.005*	.05	5.9	9.9	10.7
.005	.005	7.2	12.6	12.2

TABLE 16

OPTIMIZATION OF THREE STATE FILTER (y COMPONENT) FOR FLIGHT PASS 12

<u>a</u>	<u>RMS (ϵ_y)</u>	<u>RMS ($\epsilon_{\dot{y}}$)</u>	<u>RMS ($\epsilon_{\ddot{y}}$)</u>
.1	5.2	8.1	6.1
.08	5.2	7.7	5.7
	5.0		
.044	5.0	6.9	4.8
.01	4.6	5.3	3.2
.005	4.5	4.7	2.8
.001*	4.1	3.9	2.2
.0005	4.0	3.6	2.0

TABLE 17

OPTIMIZATION OF FOUR STATE FILTER (y COMPONENT) FOR FLIGHT PASS 12

<u>ω_0</u>	<u>β</u>	<u>RMS (ϵ_y)</u>	<u>RMS ($\epsilon_{\dot{y}}$)</u>	<u>RMS ($\epsilon_{\ddot{y}}$)</u>
.7	.165	5.6	9.9	8.8
.7	.2	5.6	10.1	9.0
.7	.05	5.3	8.4	6.6
.7	.01	4.8	6.2	4.0
.7	.005	4.6	5.4	3.3
.7	.001	4.1	4.1	2.4
.7*	.0005	4.0	3.9	2.2
1.5	.0005	4.1	4.0	3.3
1.0	.0005	4.0	3.9	2.4
.2	.0005	4.2	4.1	2.4
.05	.0005	4.1	4.0	2.4

SECTION 14

OPTIMIZATION OF THREE STATE FILTER FOR ALL FLIGHT PASSES

Before deciding upon a final filter design, it is instructive to compare the performance of the three and four state filters for all 12 flight passes rather than a selected few. To this end, it is necessary to select a single value of " a " for the three state filter and a single set of values for w_0 and β for the four state filter to be used for all 12 flight passes. Furthermore, rather than compute the value of the state dependent R matrix at every iteration, it is desirable to replace R with a constant not only for the sake of computational simplicity, but also because it will be needed later for implementation of a steady state filter design.

In this section, the parameter " a " of the three state filter is optimized over all flight passes for each coordinate. A constant R is also chosen which optimizes filter performance. Comparison of the three and four state filters is made for constant R in the next section.

Tables 18, 19, and 20 exhibit the performance of the three state y coordinate filter for all flight passes by using the full state dependent R , as well as values of 31 and 100 for R . (Recall that for a single axis, R is a 1×1 matrix, i.e., a scalar). The number 31 was obtained by averaging over the sensor variables R, θ, ϕ . (See Appendix D). The choice for " a " is .05. Notice that, overall, the filter sensitivity to R is mild. Furthermore, $R = 100$ results in better filter performance than $R = 31$. Table 21 shows the filter performance for " a " = .1 with $R = 100$. This results in a slight overall degradation in performance. However, experimentation with various values of " a " for the x axis indicated that " a " = .1 results in significant improvement in the x coordinate filter over the value of " a " = .05. This improvement more than compensates for the slight degradation in performance of the y filter for " a " = .1 over " a " = .05. Thus, the value of " a " is chosen as .1 for both the x and y coordinate filters. The performance of the x coordinate filter is shown in Table 22.

Although there is no a-priori reason for the optimal value of " a " for the z filter to be the same as for the x and y filters, the results of Tables 23, 24 and 25 indicate that the value of .1 and " a " is also optimal for the z filter.

The performance of the three state filter is illustrated graphically in Figures 5a through 5c in which the x component of target position, velocity and acceleration is plotted against time, together with the filtered estimates of these quantities. This is done for flight pass 1.

The value of " a " which parameterizes the three state filter is .1. Figures 6a through 6c are the corresponding error curves. Of particular interest is Figure 5c in which the true and filtered acceleration curves are contrasted. These curves typify the lag associated with the filtered acceleration estimates. The degree to which the filtered acceleration lags behind the true acceleration is generally influenced by the intensity of the white

TABLE 18

PERFORMANCE OF 3 STATE, y COORDINATE FILTER FOR "a" = .05, \underline{R} = STATE DEPENDENT

<u>Flight Pass</u>	<u>RMS ϵ_y</u>	<u>RMS $\epsilon_{\dot{y}}$</u>	<u>RMS $\epsilon_{\ddot{y}}$</u>
1	4.2	8.2	9.2
2	5.0	8.6	7.4
3	4.9	8.8	9.7
4	4.1	8.2	9.7
5	5.2	9.4	9.8
6	4.9	9.4	9.8
7	4.8	8.9	9.7
8	4.5	9.1	13.2
9	5.7	10.6	11.6
10	5.5	9.6	10.3
11	5.7	10.0	10.6
12	5.7	9.0	6.9
Average	5.0	9.2	9.8

TABLE 19

PERFORMANCE OF 3 STATE, y COORDINATE FILTER FOR "a" = .05, \underline{R} = 31

<u>Flight Pass</u>	<u>RMS ϵ_y</u>	<u>RMS $\epsilon_{\dot{y}}$</u>	<u>RMS $\epsilon_{\ddot{y}}$</u>
1	4.2	8.4	9.4
2	5.1	9.3	8.3
3	5.0	9.2	10.2
4	4.2	8.4	9.9
5	5.4	9.9	10.5
6	4.9	9.5	10.2

TABLE 19 (Cont)

<u>Flight Pass</u>	<u>RMS ϵ_y</u>	<u>RMS $\epsilon_{\dot{y}}$</u>	<u>RMS $\epsilon_{\ddot{y}}$</u>
7	4.8	9.0	9.9
8	4.5	9.4	13.5
9	5.9	11.2	12.2
10	5.7	10.3	11.2
11	5.7	10.8	11.4
12	5.9	10.2	8.6
Average	5.1	9.6	10.4

TABLE 20

PERFORMANCE OF 3 STATE, y COORDINATE FILTER FOR " a " = .05, R = 100

<u>Flight Pass</u>	<u>RMS ϵ_y</u>	<u>RMS $\epsilon_{\dot{y}}$</u>	<u>RMS $\epsilon_{\ddot{y}}$</u>
1	3.9	7.4	8.4
2	4.8	7.4	5.9
3	4.9	8.6	9.8
4	4.1	7.8	8.9
5	5.2	8.8	9.1
6	4.5	8.1	8.6
7	4.6	8.2	9.5
8	4.6	10.0	14.4
9	5.8	10.4	11.2
10	5.4	9.4	10.6
11	5.7	9.8	10.3
12	5.6	8.0	5.5
Average	4.9	8.7	9.4

TABLE 21

PERFORMANCE OF 3 STATE, y COORDINATE FILTER FOR "a" = .1, \underline{R} = 100

<u>Flight Pass</u>	<u>RMS ϵ_y</u>	<u>RMS $\epsilon_{\dot{y}}$</u>	<u>RMS $\epsilon_{\ddot{y}}$</u>
1	4.0	7.7	8.6
2	4.9	8.2	6.7
3	4.9	8.6	9.4
4	4.1	7.9	9.2
5	5.2	9.1	9.3
6	4.7	8.5	8.9
7	4.7	8.3	9.2
8	4.4	9.2	13.3
9	5.7	10.5	11.3
10	5.5	9.5	10.3
11	5.7	10.0	10.3
12	5.7	8.9	6.6
Average	5.0	8.9	9.4

TABLE 22

PERFORMANCE OF 3 STATE, x COORDINATE FILTER FOR "a" = .1, \underline{R} = 100

<u>Flight Pass</u>	<u>RMS ϵ_x</u>	<u>RMS $\epsilon_{\dot{x}}$</u>	<u>RMS $\epsilon_{\ddot{x}}$</u>
1	2.8	6.3	8.9
2	3.2	6.8	9.5
3	3.2	9.5	15.2
4	3.0	6.6	9.6
5	2.4	6.2	9.5
6	3.3	9.8	15.7

TABLE 22 (Cont)

<u>Flight Pass</u>	<u>RMS ϵ_x</u>	<u>RMS ϵ'_x</u>	<u>RMS ϵ''_x</u>
7	3.0	7.5	10.9
8	3.3	9.2	14.0
9	3.3	8.8	13.8
10	3.9	9.8	15.1
11	4.0	10.6	16.4
12	3.7	6.2	5.6
Average	3.7	8.1	12.0

TABLE 23

PERFORMANCE OF 3 STATE z COORDINATE FILTER FOR "a" = .01, \underline{R} = 100

<u>Flight Pass</u>	<u>RMS ϵ_z</u>	<u>RMS ϵ'_z</u>	<u>RMS ϵ''_z</u>
1	4.9	8.6	8.5
2	5.9	12.8	13.9
3	5.1	8.5	8.1
4	4.6	7.6	8.5
5	4.0	7.5	8.7
6	5.6	12.1	13.6
7	5.3	10.7	11.1
8	5.6	10.7	11.1
9	5.7	13.4	15.6
10	5.5	12.9	14.8
11	5.4	12.5	13.9
12	3.7	6.6	5.8
Average	5.1	10.3	11.1

TABLE 24

PERFORMANCE OF 3 STATE, z COORDINATE FILTER FOR "a" = .05, $\underline{R} = 100$

<u>Flight Pass</u>	<u>RMS ϵ_z</u>	<u>RMS ϵ'_z</u>	<u>RMS ϵ''_z</u>
1	4.5	7.7	7.8
2	4.5	9.5	11.7
3	4.7	7.9	7.7
4	4.2	6.9	7.6
5	4.1	7.0	8.2
6	4.4	9.3	12.0
7	4.4	8.9	10.3
8	4.8	9.5	10.7
9	4.1	9.8	13.4
10	4.0	9.2	12.3
11	4.1	9.4	12.0
12	4.0	7.0	6.1
Average	4.3	8.5	10.0

TABLE 25

PERFORMANCE OF 3 STATE, z COORDINATE FILTER FOR "a" = .1, $\underline{R} = 100$

<u>Flight Pass</u>	<u>RMS ϵ_z</u>	<u>RMS ϵ'_z</u>	<u>RMS ϵ''_z</u>
1	4.6	8.0	7.9
2	4.2	8.8	10.9
3	4.7	8.2	7.9
4	4.4	7.2	7.7
5	4.2	7.3	8.2
6	4.2	8.7	11.4

TABLE 25 (Cont)

<u>Flight Pass</u>	<u>RMS ϵ_z</u>	<u>RMS ϵ'_z</u>	<u>RMS ϵ''_z</u>
7	4.2	8.7	10.1
8	4.7	9.4	10.8
9	3.7	9.0	12.7
10	3.7	8.5	11.6
11	4.0	8.8	11.3
12	4.1	7.5	6.5
Average	4.2	13.8	9.8

noise driving the acceleration process as well as the number of derivatives needed to obtain the acceleration estimate from the raw sensor data (in this case, two).

In summary, we achieved a rather simple filter design in which three identical and independent filters operate for each coordinate with a constant R of 100 and the parameter "a" having a value of .1.

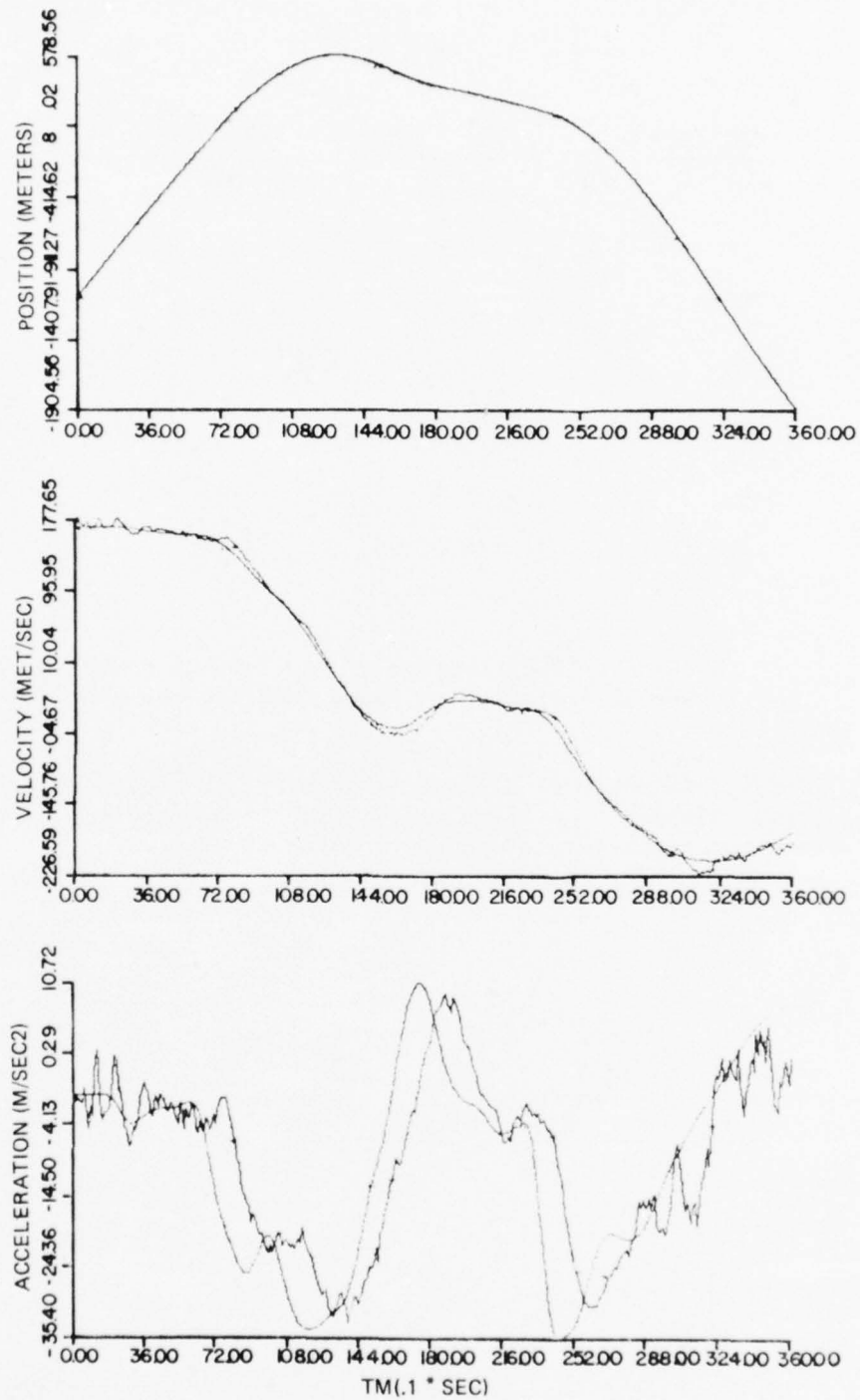


Figure 5. Filtered and true position, velocity, and acceleration profiles along x for flight pass 1

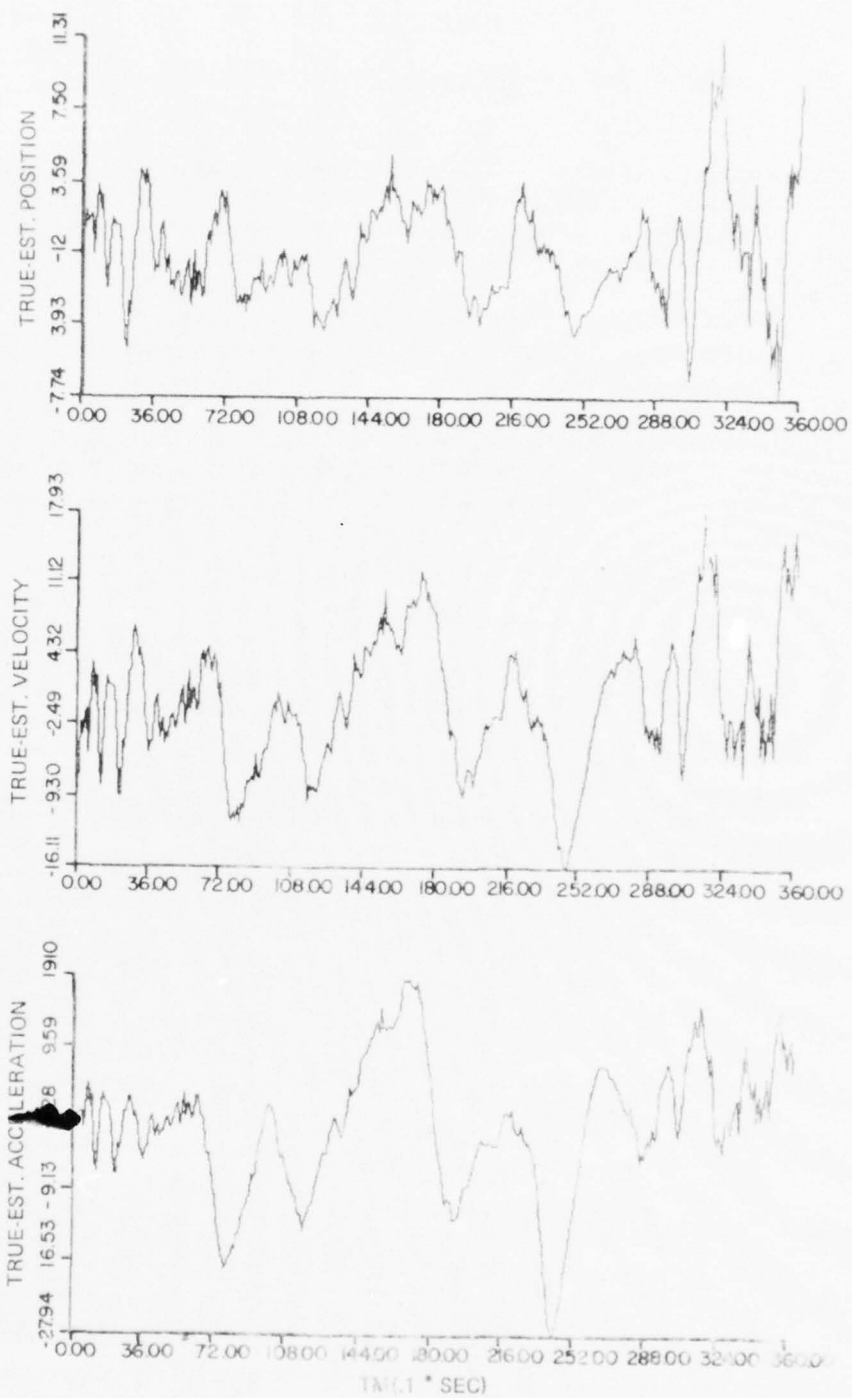


Figure 6. State estimation error curves for x component of target position, velocity, and acceleration for flight pass 1

SECTION 15

COMPARISON OF THE THREE AND FOUR STATE FILTERS

In this section, a final comparison of the three and four state filters is made. With the view that a decision based on 12 flight passes is better than one based on three, the determination of the more desirable filter design is made using all 12 available flight passes. Again, as in the case of the three state filter, a choice is made for parameters ω_0 and β as well as for R . With $\omega_0 = .1$ and $\beta = .03$, Tables 26, 27, and 28 indicate that the appropriate choice for R is again 100. Comparing Tables 21 of Section 14 with Table 28, we see that the performance of the two filters is virtually identical. Thus, the choice of one filter over another reduces to a choice between a simple filter design and a slightly more complex one. On the basis of these arguments, as well as the results of the foregoing sections, we conclude that the three state filter offers the best features for implementation in a fire control system.

TABLE 26

PERFORMANCE OF 4 STATE, y COORDINATE FILTER FOR
 $\omega_0 = .1$, $\beta = .03$, $R =$ STATE DEPENDENT

Flight Pass	RMS ϵ_y	RMS $\epsilon_{\dot{y}}$	RMS $\epsilon_{\ddot{y}}$
1	4.1	8.0	9.1
2	4.9	8.2	7.1
3	5.0	8.8	10.1
4	4.1	8.1	9.6
5	5.2	9.2	9.8
6	4.8	9.0	9.7
7	4.8	8.7	9.9
8	4.5	9.4	13.9
9	5.7	10.6	11.7
10	5.5	9.5	10.7
11	5.6	10.0	11.0
12	5.7	8.6	6.5
Average	5.0	9.0	9.9

TABLE 27

PERFORMANCE OF 4 STATE, y COORDINATE FILTER FOR
 $\omega_0 = .1$, $\beta = .03$, $\underline{R} = 31$

<u>Flight Pass</u>	<u>RMS ϵ_y</u>	<u>RMS $\epsilon_{\dot{y}}$</u>	<u>RMS $\epsilon_{\ddot{y}}$</u>
1	4.1	8.1	9.3
2	5.1	8.8	7.9
3	5.1	9.2	10.6
4	4.2	8.4	9.8
5	5.3	9.7	10.4
6	4.8	9.1	9.9
7	4.8	8.8	10.0
8	4.6	9.7	14.3
9	5.9	11.1	12.3
10	5.7	10.2	11.4
11	5.7	10.7	11.7
12	5.8	9.7	8.1
Average	5.1	9.5	10.5

TABLE 28

PERFORMANCE OF 4 STATE, y COORDINATE FILTER FOR
 $\omega_0 = .1$, $\beta = .03$, $\underline{R} = 100$

<u>Flight Pass</u>	<u>RMS ϵ_y</u>	<u>RMS $\epsilon_{\dot{y}}$</u>	<u>RMS $\epsilon_{\ddot{y}}$</u>
1	3.9	7.5	8.5
2	4.7	7.2	5.8
3	5.0	9.1	10.4
4	4.2	7.9	8.9

TABLE 28 (Cont)

<u>Flight Pass</u>	<u>RMS ϵ_y</u>	<u>RMS $\epsilon_{\dot{y}}$</u>	<u>RMS $\epsilon_{\ddot{y}}$</u>
5	5.2	8.8	9.1
6	4.4	8.1	8.8
7	4.7	8.6	10.2
8	4.9	11.0	15.4
9	5.9	10.6	11.5
10	5.5	9.7	11.2
11	5.7	10.1	10.9
12	5.5	7.7	5.2
Average	5.0	8.9	9.7

SECTION 16

STEADY STATE FILTER

The steady state filter offers the simplest filter design considered thus far. This filter is implemented by replacing the elements of the gain matrix with its steady state values. For the three state filter, the gain \underline{K} is a three component column vector. Its steady state value is determined by running the filter over an entire flight path. For the three state filter with " a " = .1 and \underline{R} = 100, the steady state gains are $K_p = .27$, $K_v = .34$, $K_a = .44$.

Tables 29, 30, 31 illustrate the performance of the steady state filter. Table 32 is a composite of Tables 29 through 31 where the RMS errors along x, y, and z are combined to give the total position, velocity, and acceleration RMS errors in the state estimates. Table 33 is the analogue of Table 32 for the non-steady state filter. Comparison of Table 32 with Table 33 (or Tables 29 through 31 with Tables 21, 22, and 25), indicate that the steady state filter performs almost as well as the non-steady state design. Indeed, the difference in performance of the two filters may be attributed to the transient response of the time varying gains in the non-steady state filter.

TABLE 29

PERFORMANCE OF 3 STATE, x COORDINATE STEADY STATE FILTER

Flight Pass	RMS ϵ_x	RMS ϵ'_x	RMS ϵ''_x
1	2.8	7.0	9.7
2	3.2	7.3	10.0
3	3.3	9.8	15.1
4	3.0	6.9	9.8
5	2.4	6.7	9.5
6	3.5	10.5	16.1
7	3.2	8.1	11.5
8	3.5	9.5	14.4
9	3.3	9.3	14.7
10	4.0	10.4	15.5
11	4.0	11.2	17.0

TABLE 29 (Cont)

<u>Flight Pass</u>	<u>RMS ϵ_x</u>	<u>RMS ϵ_x^{\cdot}</u>	<u>RMS $\epsilon_x^{\cdot\cdot}$</u>
12	3.7	7.2	7.0
Average	3.3	8.7	12.5

TABLE 30

PERFORMANCE OF 3 STATE y COORDINATE STEADY STATE FILTER

<u>Flight Pass</u>	<u>RMS ϵ_y</u>	<u>RMS ϵ_y^{\cdot}</u>	<u>RMS $\epsilon_y^{\cdot\cdot}$</u>
1	4.0	8.5	10.2
2	4.9	9.3	9.4
3	4.9	9.7	11.4
4	4.1	8.9	11.3
5	5.2	10.2	11.9
6	4.6	11.5	11.6
7	4.6	9.0	11.0
8	4.4	10.0	15.0
9	5.7	11.4	13.2
10	5.5	10.1	11.9
11	5.7	10.6	11.9
12	5.7	9.7	8.9
Average	4.9	9.9	11.5

Figure 7 is a plot of K_p , K_v , and K_a for the three state, y component filter with "a" = .05. (This is the optimal value of "a" for flight pass 1 for which these plots were made). The conclusion that can be drawn from these curves is significant. Namely, that after about 3 seconds, during which the gains settle to their steady state values, the performance of the two filters is in essence identical.

TABLE 31

PERFORMANCE OF 3 STATE z COORDINATE STEADY STATE FILTER

<u>Flight Pass</u>	<u>RMS ϵ_z</u>	<u>RMS $\epsilon_{\dot{z}}$</u>	<u>RMS $\epsilon_{\ddot{z}}$</u>
1	4.6	8.0	8.5
2	4.2	8.6	10.9
3	4.7	8.1	8.6
4	4.4	7.3	8.5
5	4.2	7.1	8.4
6	4.2	8.7	12.0
7	4.2	8.6	10.7
8	4.7	9.5	11.8
9	3.7	8.9	12.7
10	3.7	8.4	11.6
11	4.0	8.8	11.4
12	4.0	7.2	6.4
Average	4.2	8.3	10.1

TABLE 32

TOTAL RMS ERRORS FOR 3 STATE, STEADY STATE FILTERS

<u>Flight Pass</u>	<u>RMS ϵ_{pos}</u>	<u>RMS ϵ_{vel}</u>	<u>RMS ϵ_{acc}</u>
1	6.7	13.6	16.5
2	7.2	14.6	17.5
3	7.5	16.0	20.8
4	6.7	13.4	17.2
5	7.1	14.1	17.4

TABLE 32 (Cont)

<u>Flight Pass</u>	<u>RMS ϵ_{pos}</u>	<u>RMS ϵ_{vel}</u>	<u>RMS ϵ_{acc}</u>
6	7.1	17.9	23.2
7	7.0	14.9	19.2
8	7.2	16.8	23.8
9	7.6	17.2	23.5
10	7.7	16.8	22.7
11	8.0	17.7	23.6
12	7.9	14.1	13.0
Average	7.3	15.6	19.9

TABLE 33

TOTAL RMS ERRORS FOR 3 STATE FILTERS WITH $\underline{R} = 100$

<u>Flight Pass</u>	<u>RMS ϵ_{pos}</u>	<u>RMS ϵ_{vel}</u>	<u>RMS ϵ_{acc}</u>
1	6.7	12.8	14.7
2	7.2	13.8	15.9
3	7.5	15.2	19.6
4	6.7	12.6	15.4
5	7.1	13.2	15.6
6	7.1	15.7	21.3
7	7.0	14.2	17.5
8	7.2	16.1	22.2
9	7.6	16.4	21.9
10	7.7	16.1	21.6
11	8.0	17.0	22.4

TABLE 33 (Cont)

<u>Flight Pass</u>	<u>RMS ϵ_{pos}</u>	<u>RMS ϵ_{vel}</u>	<u>RMS ϵ_{acc}</u>
12	8.0	13.2	10.8
Average	7.3	14.7	18.3

Figures 8 through 13 are typical error curves for both filters. As expected there is little difference between the two sets of curves. With the exception of occasional peaked values after settling has occurred, the more sizeable errors occur early in time.

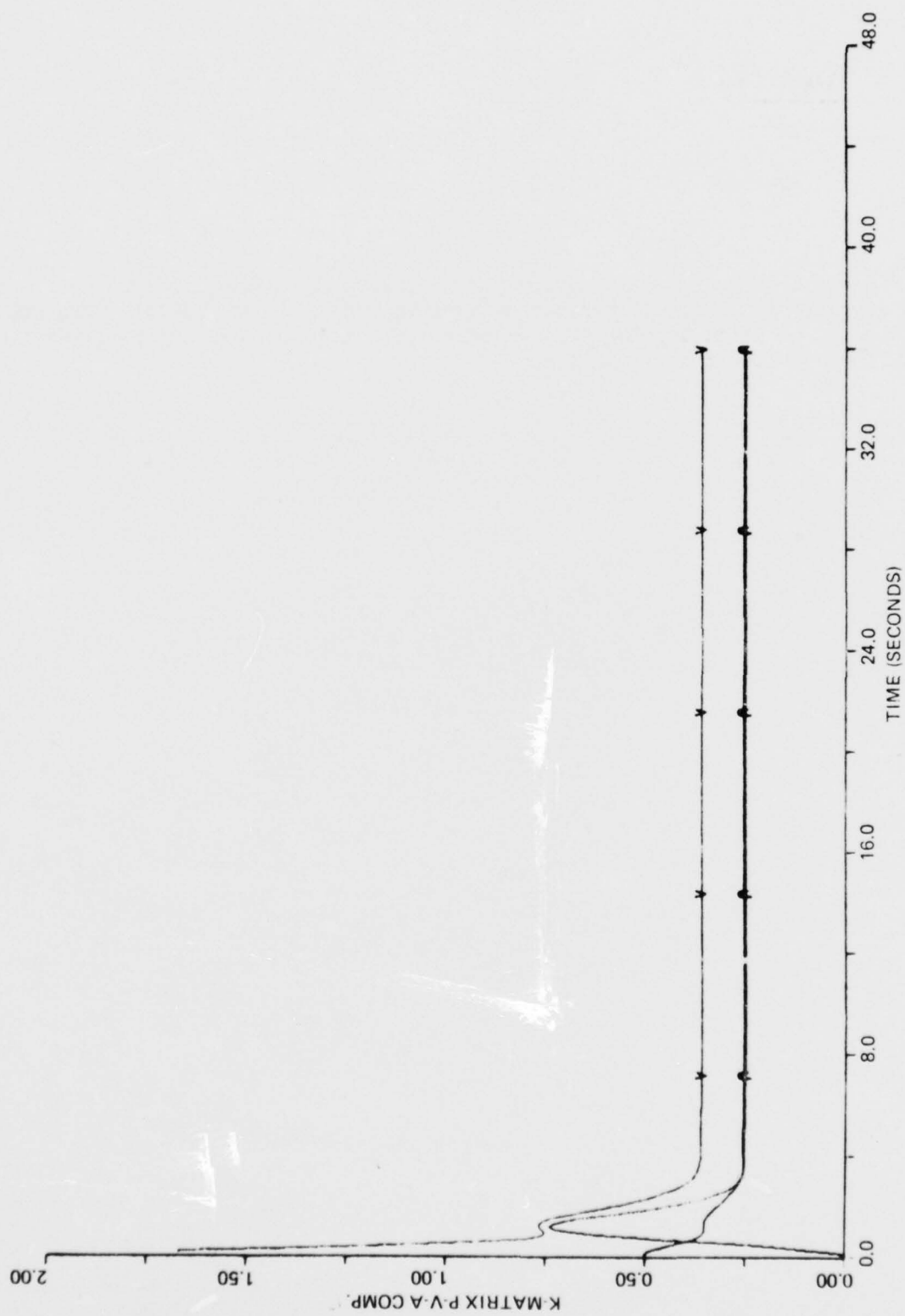


Figure 7. Position, velocity, and acceleration gains for the three state filter

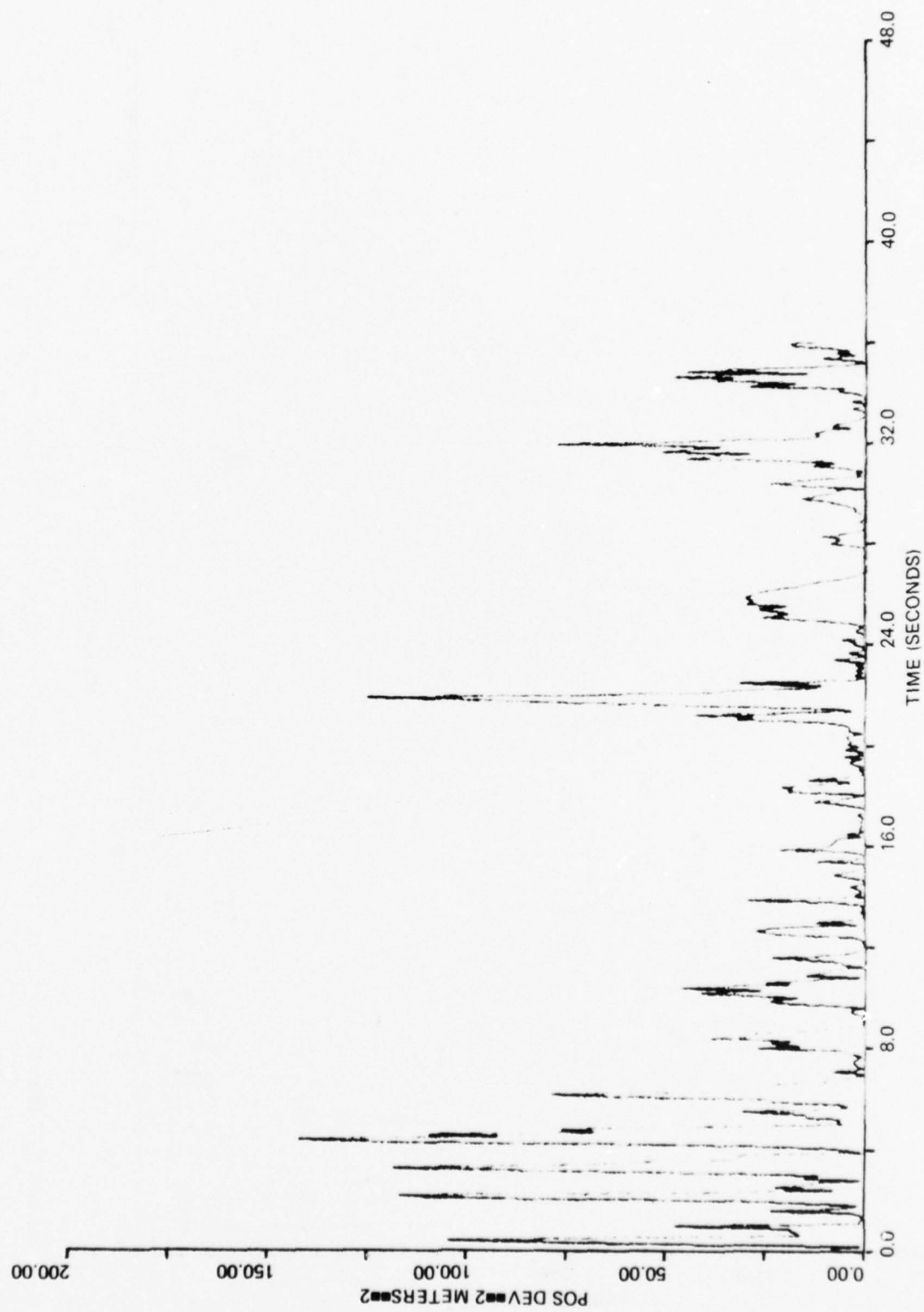


Figure 8. Error curve for state estimates of y component of target position for flight pass 1

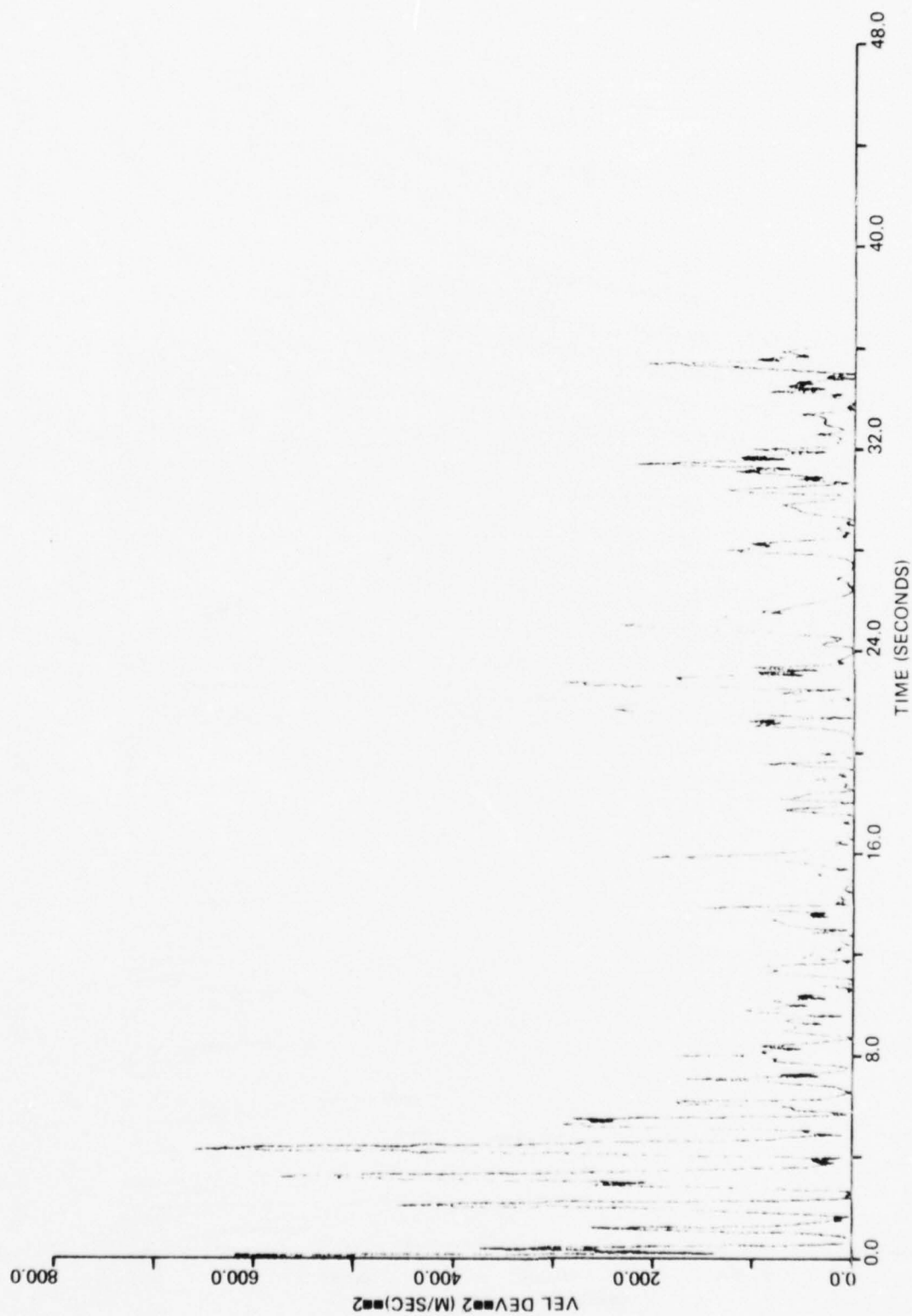


Figure 9. Error curve for state estimates of y component of target velocity for flight pass 1

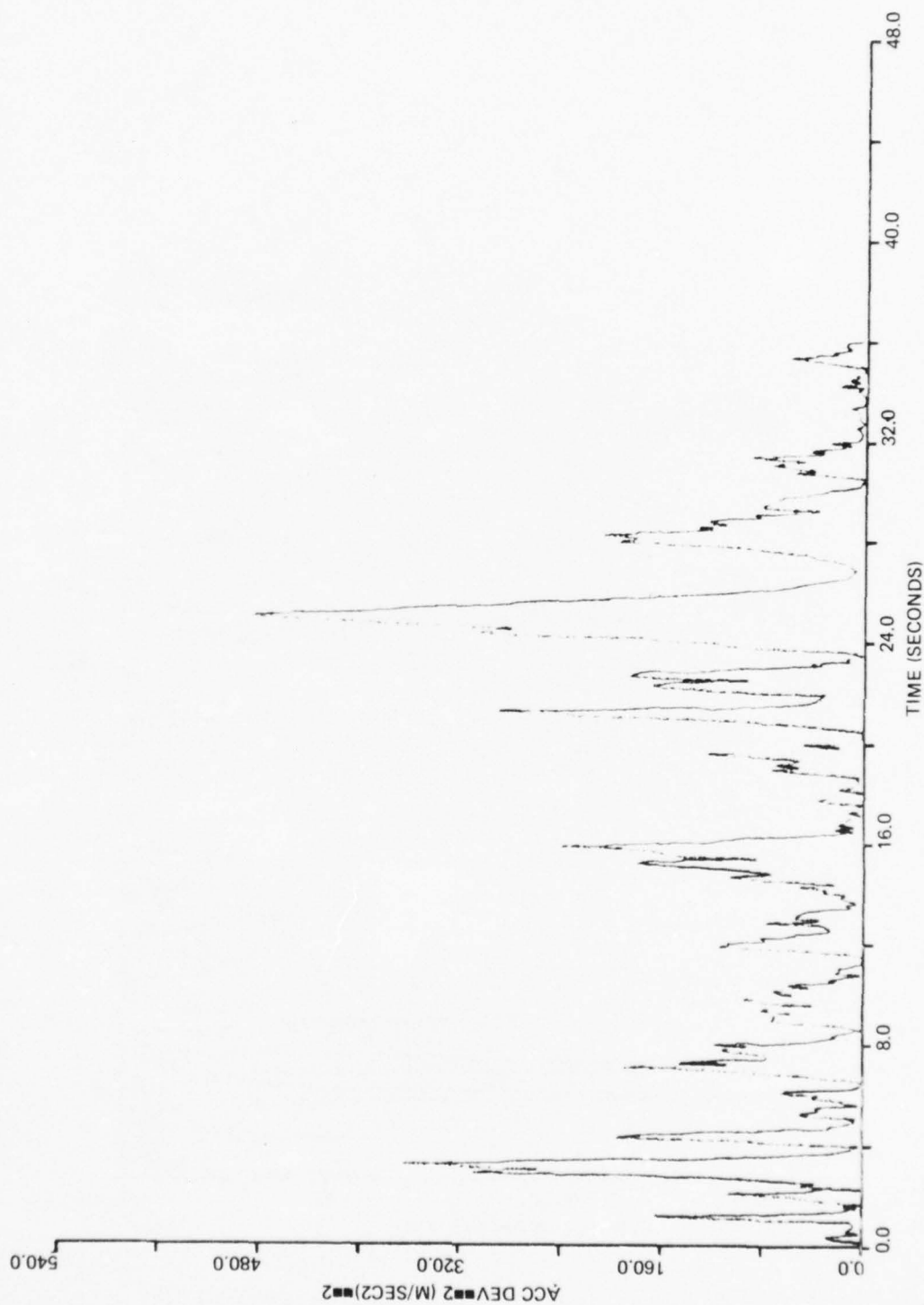


Figure 10. Error curve for state estimates of y component of target acceleration for flight pass 1.

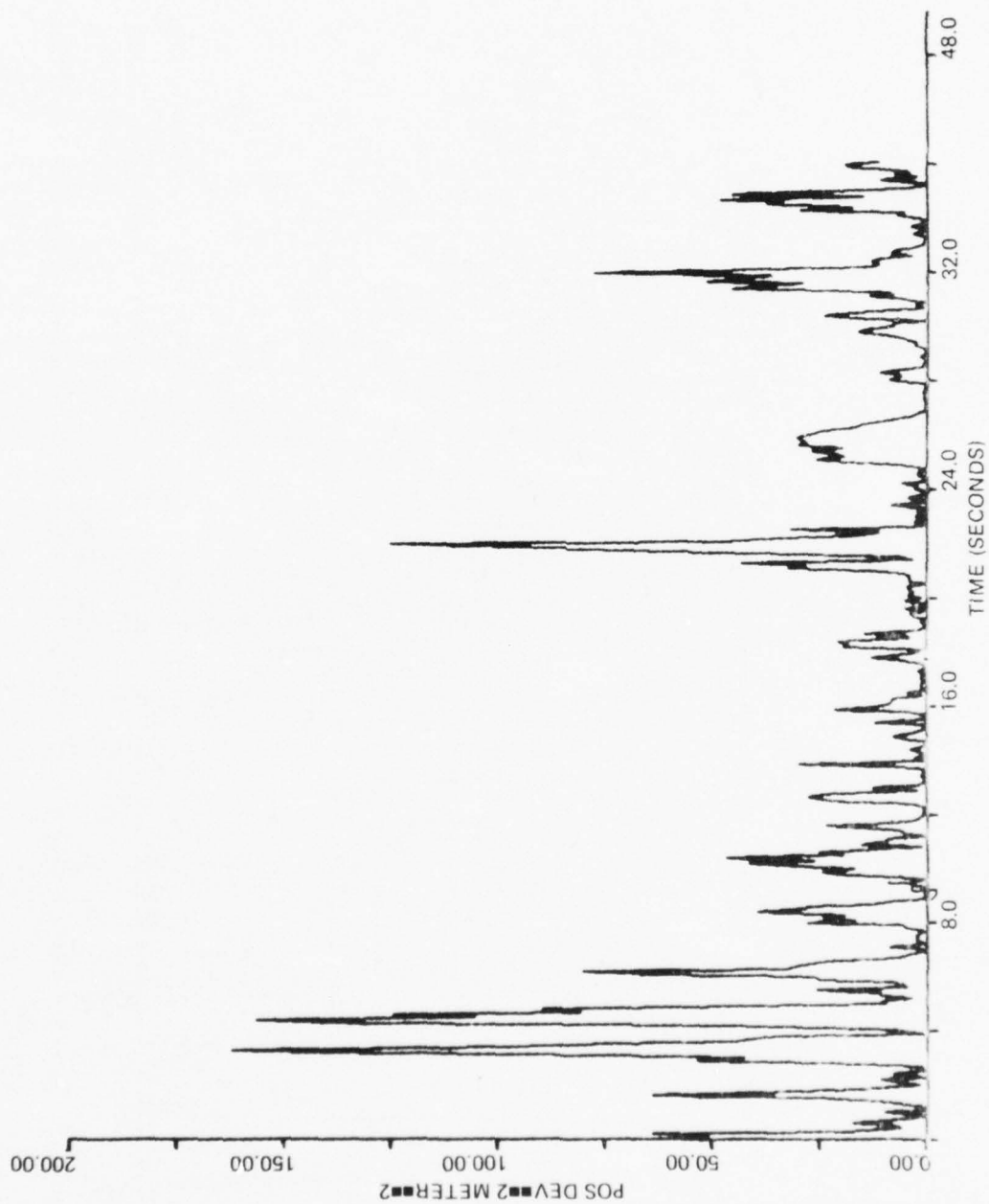


Figure 11. Error curve generated with steady state filter for state estimates of y component of target position for flight pass 1

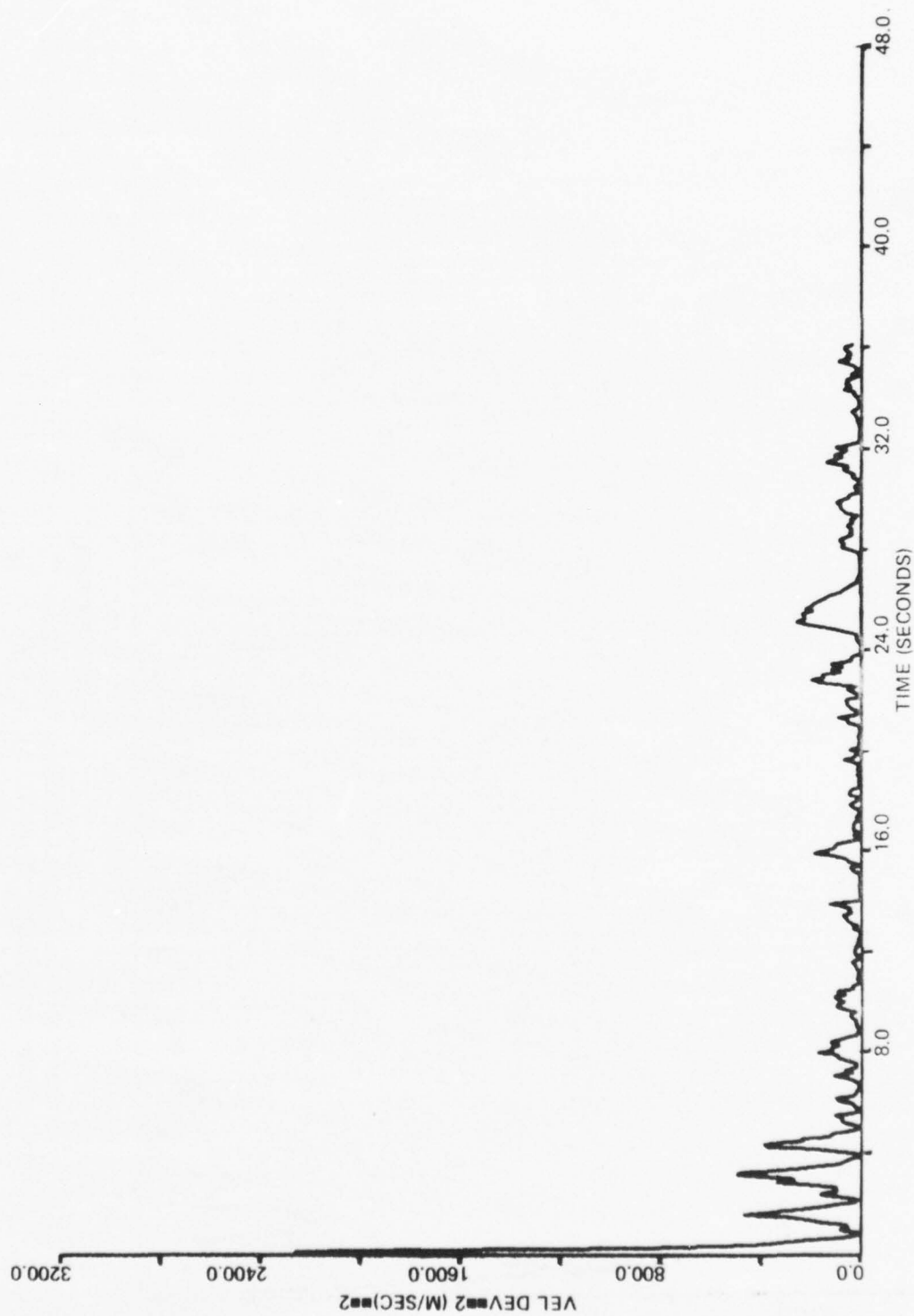


Figure 12. Error curve generated with steady state filter for state estimates of y component of target velocity for flight pass 1

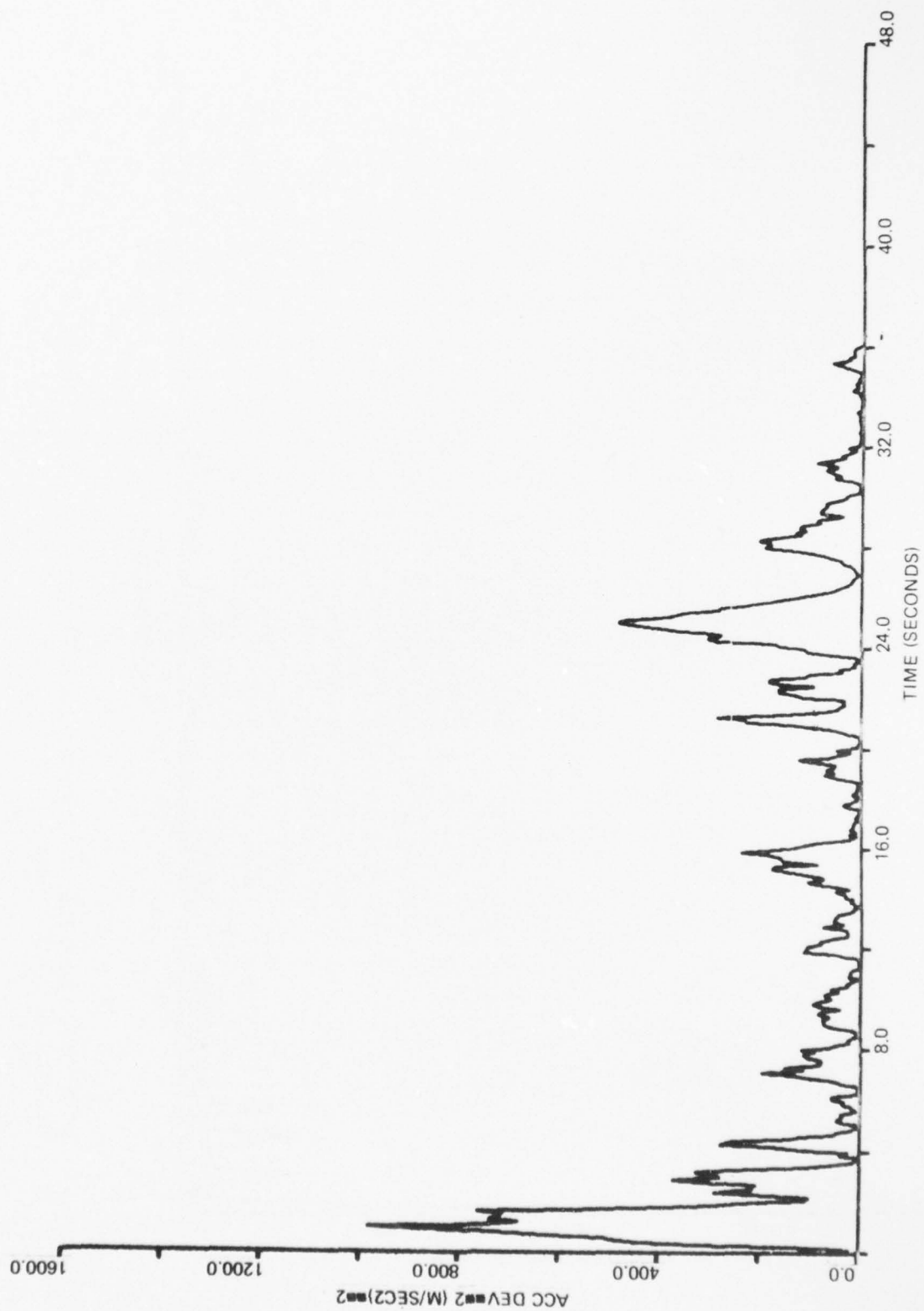


Figure 13. Error curve generated with steady state filter for state estimates of y component of target acceleration for flight pass 1

SECTION 17

IMPROVED SENSOR

Thus far, filter performance was judged relative to a sensor which is characterized by a standard deviation in range of 10m and a $1\sqrt{2}$ standard deviation in both azimuth and elevation measurements. In this section, the increase in estimation accuracy achieved when a higher quality sensor is employed is investigated. Here, the sensor is assumed to have a standard deviation in range of 5m, and a $1/2\sqrt{2}$ standard deviation in both azimuth and elevation. Thus, the increase in sensor accuracy is twofold.

The sensor model must now be modified to accommodate the change in sensor accuracy. This is accomplished by changing the value of R from 100 to 25.

Table 34 lists the total RMS position, velocity, and acceleration estimation errors over all three coordinate axes, for all 12 flight passes. The results indicate that an improvement in the raw sensor data does not result in a corresponding improvement in filtered state estimates. Whereas the raw sensor measurements are increased in accuracy by a factor of two, the average improvement in the position estimate is 1.9 to 1, while the velocity and acceleration estimates improve by factors of 3 to 2 and 6 to 5 respectively.

In conclusion, doubling the sensor accuracy does not double the state estimation accuracy, and the relative improvement in state estimation decreases as one goes from velocity to acceleration estimates.

TABLE 34

TOTAL RMS ERRORS FOR 3 STATE FILTER WITH IMPROVED SENSOR

<u>Flight Pass</u>	<u>RMS ϵ_{pos}</u>	<u>RMS ϵ_{vel}</u>	<u>RMS ϵ_{acc}</u>
1	3.6	8.7	12.6
2	3.9	9.1	13.3
3	4.0	10.0	13.1
4	3.6	9.6	13.4
5	3.9	8.9	13.7
6	3.9	10.4	18.1
7	3.7	9.3	14.7
8	3.9	10.8	19.1

TABLE 34 (Cont)

<u>Flight Pass</u>	<u>RMS ϵ_{pos}</u>	<u>RMS ϵ_{vel}</u>	<u>RMS ϵ_{acc}</u>
9	4.1	10.8	18.5
10	4.1	10.6	18.1
11	4.1	11.1	18.7
12	4.2	8.9	15.2
Average	3.9	9.9	15.2

SECTION 18

RATE AIDED FILTER

As a rule, the accuracy of state estimation increases with an increase in the number of independent sensor measurements. Present day fire control concepts as well as operational systems such as GLAADS make use of rate data for state estimation. However, the sensor measurements are not necessarily all independent. Range rate, for example, is usually derived by processing radar return signals. The manner of processing is also germane to the radar unit. The increase in filter performance given rate inputs for a hypothetical position and rate sensor is determined in this section. The sensor model is hypothetical because sufficient data is, at present, lacking for the construction of an accurate sensor model. The sensor is therefore modeled as having position and rate measurements which are uncorrelated with each other.

The position measurements are assumed to be of the same quality as in earlier sections. That is, the standard deviation of the range error is 10m while the angular errors have a standard deviation of 1m. Range rate error is assumed to have zero mean and standard deviation of 5 m/sec, while angular rate errors are characterized by a standard deviation of 1/2m/sec with zero mean.

The modification in the three state filter with these additional measurements is minor. The \underline{H} matrix in the sensor model $\underline{Z} = \underline{H}\underline{x} + \underline{D}\underline{v}$ is modified to

$$(18-1) \quad \underline{H} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

whereas \underline{v} is now a two element column vector,

$$(18-2) \quad \underline{v} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

As a consequence, the error covariance matrix \underline{R} for the observation model is a 2 x 2 matrix with (Appendix D)

$$(18-3) \quad R_{11} = R_\theta^2 [\sigma_\theta^2 \cos^2 \theta \cos^2 \varphi + \sigma_\theta^2 \sigma_\varphi^2 \cos^2 \theta \sin^2 \varphi + \sigma_\varphi^2 \sin^2 \theta \sin^2 \varphi] + \sigma_R^2 \sin^2 \theta \cos^2 \varphi$$

$$(18-4) \quad R_{12} = R_{21} = R_\theta \dot{R} (\sigma_\varphi^2 \sin^2 \theta \sin^2 \varphi + \sigma_\theta^2 \cos^2 \theta \cos^2 \varphi + \sigma_\theta^2 \sigma_\varphi^2 \cos^2 \theta \sin^2 \varphi) - \dot{\theta} \sigma_\varphi^2 \varphi^2 \sin \theta \cos \theta \sin^2 \varphi - R \dot{\varphi} \sigma_\theta^2 \cos^2 \theta \sin \varphi \cos \varphi$$

$$\begin{aligned}
(18-5) \quad R_{22} = & (\dot{R}^2 + \sigma_R^2) [\sigma_\varphi^2 \sin^2 \theta \sin^2 \varphi + \sigma_\theta^2 \cos^2 \theta \cos^2 \varphi \\
& + \sigma_\theta^2 \sigma_\varphi^2 \cos^2 \theta \sin^2 \varphi] + \sigma_R^2 \sin^2 \theta \cos^2 \varphi \\
& + R^2 (\dot{\sigma}^2 + \sigma_\theta^2) [\sigma_\varphi^2 \cos^2 \theta \sin^2 \varphi + \sigma_\theta^2 \sin^2 \theta \cos^2 \varphi \\
& + \sigma_\theta^2 \sigma_\varphi^2 \sin^2 \theta \sin^2 \varphi] + R^2 \sigma_\theta^2 \cos^2 \theta \cos^2 \varphi \\
& + R^2 (\varphi^2 + \sigma_\varphi^2) [\sigma_\varphi^2 \sin^2 \theta \cos^2 \varphi + \sigma_\theta^2 \cos^2 \theta \sin^2 \varphi \\
& + \sigma_\theta^2 \sigma_\varphi^2 \cos^2 \theta \cos^2 \varphi] + R^2 \sigma_\varphi^2 \sin^2 \theta \sin^2 \varphi \\
& + 2 R \dot{R} (\dot{\varphi} \sigma_\varphi^2 \sin^2 \theta \cos \varphi + \dot{\theta} \sigma_\varphi^2 \sin \theta \cos \theta \sin \varphi \\
& - \sigma_\theta^2 \dot{\varphi} \cos^2 \theta \cos \varphi) \sin \varphi
\end{aligned}$$

for the x coordinate filter. The elements of the \underline{R} matrix for the y coordinate filter are obtained by simply replacing by $90^\circ - \varphi$ in the above expressions. For the z coordinate filter, the elements of \underline{R} are

$$(18-6) \quad R_{11} = R^2 \sigma_\theta^2 \sin^2 \theta + \sigma_R^2 \cos^2 \theta$$

$$(18-7) \quad R_{12} = R_{21} = R \dot{R} \sigma_\theta^2 \sin^2 \theta$$

$$\begin{aligned}
(18-8) \quad R_{22} = & R^2 \sigma_\theta^2 \sin^2 \theta + R^2 (\dot{\theta}^2 + \sigma_\theta^2) \sigma_\theta^2 \cos^2 \theta + \sigma_R^2 \theta^2 \sin^2 \theta \\
& + (\dot{R} + \sigma_R^2) \sigma_\theta^2 \sin^2 \theta + \sigma_R^2 \cos^2 \theta + 2 R \dot{R} \dot{\theta} \sigma_\theta^2 \sin \theta \cos \theta
\end{aligned}$$

Rather than use the state dependent \underline{R} matrices above, which are after all, rather complicated, the elements of \underline{R} will be replaced by constant values. This procedure, as was demonstrated earlier, works quite well for the "position only" filter. The validity of this approach for the rate aided filter was nevertheless confirmed by implementing the y coordinate filter with the state dependent \underline{R} . The results confirmed the suspicion that there is a negligible difference in performance between the two filters. The constant \underline{R} for all three filters is chosen to be

$$(18-9) \quad \underline{R} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ & 0 & & 25 \end{bmatrix}$$

Tables 35 and 36 illustrate the performance of the rate aided and the steady state rate aided filters respectively. As seen from these tables, both filters perform quite well with marginal differences in performance.

Comparing the results of these tables with the results of Tables 33 and 34 for the "position only" filter and the filter operating with an improved sensor, the following conclusions can be drawn: On the average

1) the addition of rate information to the observations, i.e., \dot{R} , $\dot{\theta}$, $\dot{\phi}$, without changing the quality of the position measurements results in an overall increase in state estimation accuracy by a factor of 2.3 to 1 in position, 4 to 1 in velocity, and 9 to 5 in acceleration.

TABLE 35

TOTAL RMS ERRORS FOR 3 STATE, RATE AIDED FILTERS

<u>Flight Pass</u>	<u>RMS ϵ_{pos}</u>	<u>RMS ϵ_{vel}</u>	<u>RMS ϵ_{acc}</u>
1	3.0	3.3	7.9
2	3.2	3.6	9.0
3	3.2	3.7	10.3
4	2.8	3.5	8.9
5	3.2	3.6	8.9
6	3.0	3.7	10.6
7	3.0	3.3	9.0
8	3.2	3.8	11.3
9	3.2	3.8	11.4
10	3.6	4.0	11.3
11	3.5	4.2	11.5
12	3.7	4.1	8.3
Average	3.2	3.7	9.9

TABLE 36

TOTAL RMS ERRORS FOR 3 STATE, RATE AIDED STEADY STATE FILTERS

Flight Pass	RMS ϵ_{pos}	RMS ϵ_{vel}	RMS ϵ_{acc}
1	3.5	3.3	7.7
2	3.7	3.7	8.9
3	3.9	3.7	10.2
4	3.3	3.5	8.9
5	3.8	3.6	8.8
6	3.6	3.7	10.6
7	3.5	3.3	8.9
8	3.5	3.7	11.2
9	4.0	3.9	11.3
10	4.1	4.0	11.2
11	4.4	4.2	11.4
12	4.6	4.1	8.0
Average	3.8	3.7	9.8

2) doubling the accuracy of a position only filter (by reducing the R , θ , φ σ values by $1/2$), does not increase state estimation performance as much as does the addition of rate information. That is, the rate aided sensor provides better state estimates than the improved position only sensor. These estimates are more accurate by a factor of 6 to 5 in position, 2.7 to 1 in velocity, and 3 to 2 in acceleration.

The indications are, then, that if one wishes to improve the quality of target state estimates, the effort is better spent in building a rate sensor rather than an improved position only sensor.

SECTION 19

CORRELATED NOISE SENSOR

White noise is a mathematical fiction. It does not occur in reality because real events are, in fact, time correlated and do not take on values between plus and minus infinity over arbitrarily small time intervals. Nevertheless, the concept of white noise is useful because, at least in its discretized form, it is often not a bad approximation to reality. Furthermore, it is often used as a driving function for modeling a stochastic process which is characterized by time correlated events. An appropriate model for such a process, therefore, requires some knowledge of the correlation time.

In this section, we simulate the sensor noise for each of the three position measurements as a first order Markov process characterized by a correlation time of .1 sec. The filter equations will, however, remain unchanged. That is, the filter still "thinks" that the sensor noise is uncorrelated. Whatever the filter performance in this case, it should improve if one incorporates into the filter equations the appropriate sensor model for which the sensor errors are time correlated.

The correlated sensor errors are modeled as a first order Markov process by

$$\dot{x} = -ax + b\omega$$

where ω is white noise. As seen in Section 10, the solution for the discrete form is

$$x_{n+1} = \phi(\Delta)x_n + b \int_0^{\Delta} \phi(\tau)\bar{\omega} d\tau$$

where

$$\dot{\phi} = -a\phi$$

so

$$\phi = e^{-at}$$

Thus,

$$(19-1) \quad x_{n+1} = e^{-a\Delta} x_n + \frac{1}{a}(1 - e^{-a\Delta})b\bar{\omega}_n$$

where $\frac{1}{a}$ is the correlation time and b is the intensity of the white noise

sequence. The same standard deviation values are assumed as were used throughout most of the report. That is, a 10m standard deviation in range error and 1m standard deviation in the azimuth and elevation errors.

Equation (19-1) is used to generate the sequence of measurement errors in the simulation. The results of the simulation for all 12 flight passes using the now standard three state filters is provided in Table 37. The results are remarkable but not too surprising. Filtered state estimates are considerably improved when the sensor noise is correlated. The RMS errors in position drop, on the average, by 53%, the velocity RMS error decreases by 42%, and the acceleration RMS error decreases by 19%. The reason for this improvement is that a correlated sequence is, in a gross sort of way, a smoother function of time than an uncorrelated time sequence which means that it is easier for the filter to "learn" a correlated process than an uncorrelated one.

Figures 14a, 14b, and 14c exhibit typical position, velocity and acceleration profiles along the y coordinate for flight pass 1. It is evident from Figure 14c that there is a characteristic lag in the acceleration estimate which probably accounts for a significant portion of the total RMS acceleration error. The corresponding error curves are contained in Figures 15a-15b and 15c.

TABLE 37

TOTAL RMS ERRORS FOR 3 STATE FILTERS WITH CORRELATED SENSOR NOISE INPUTS

<u>Flight Pass</u>	<u>RMS ϵ_{pos}</u>	<u>RMS ϵ_{vel}</u>	<u>RMS ϵ_{acc}</u>
1	3.0	7.5	11.9
2	3.3	7.6	12.4
3	3.3	8.9	16.0
4	3.2	7.4	12.8
5	3.2	7.5	12.8
6	3.5	9.6	17.7
7	3.3	8.4	14.1
8	3.6	10.0	18.7
9	3.6	9.9	18.1
10	3.6	9.7	17.9
11	3.6	10.2	18.5
12	3.5	5.3	6.7
Average	3.5	5.3	6.7

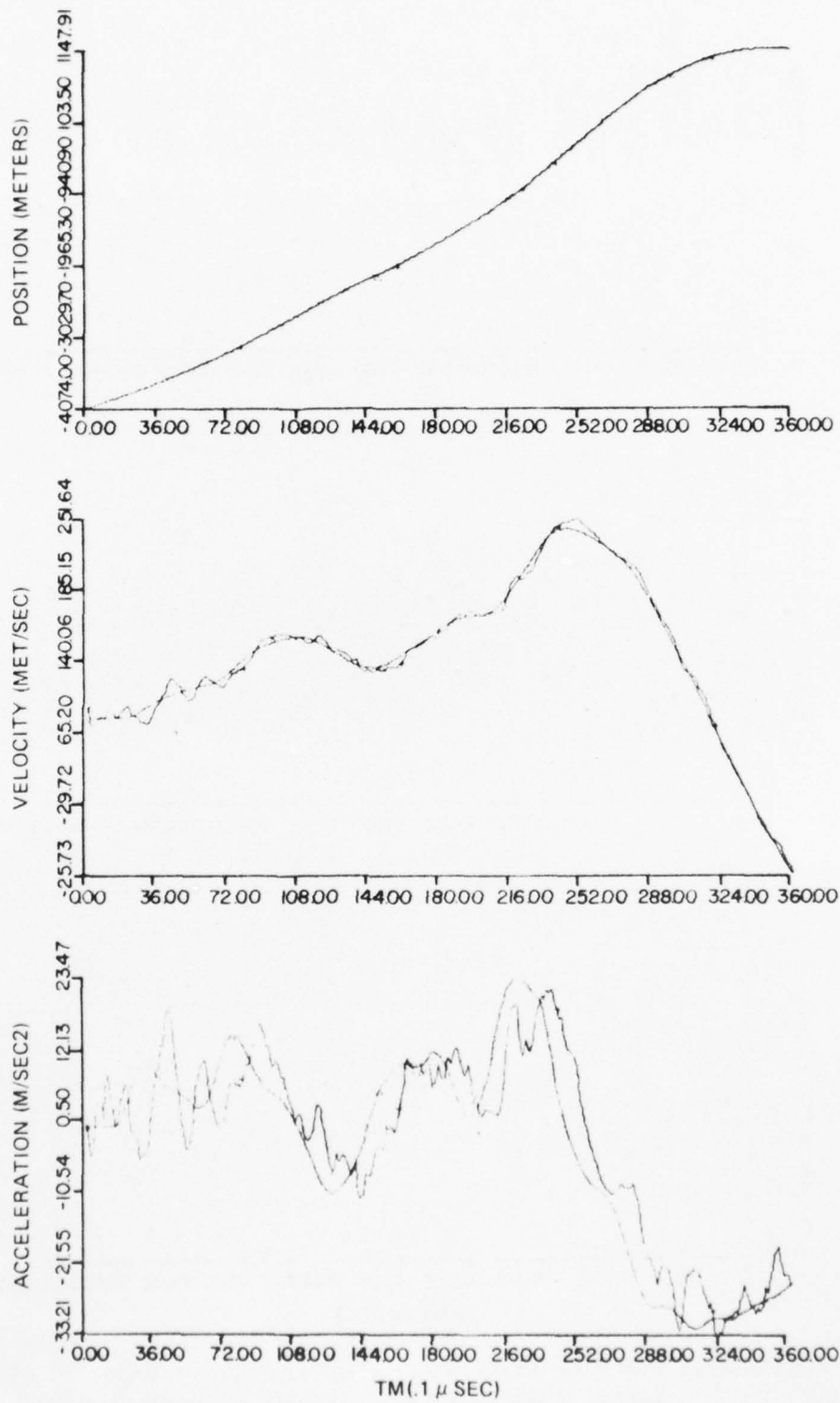


Figure 14. Filtered and true position, velocity, and acceleration profiles along y for flight pass 1

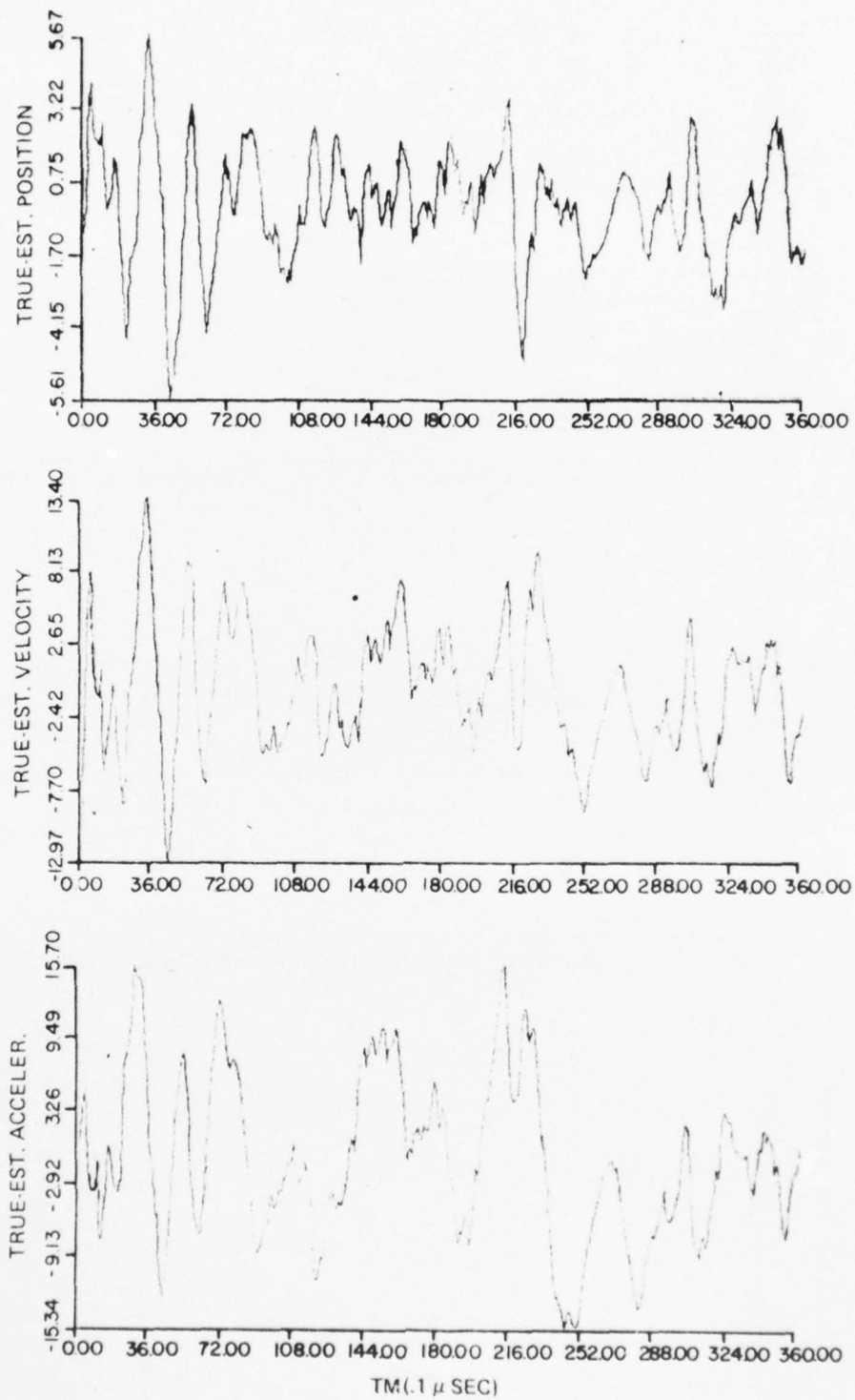


Figure 15. State estimation error curves for y component of target position, velocity, and acceleration for flight pass 1

SECTION 20

SIMULATION OF THE NINE STATE FILTER

In Section 3, we described the 9 state filter, a paradigm upon which subsequent suboptimal filter designs were based. This filter couples the state estimation errors between each of the three coordinates whereas the subsequent filters do not. One therefore expects improved performance of the 9 state filter over the 3 state filters. Solution for the state estimates using this filter, however, involves more computation because of the need to do matrix inversion at each iteration. In contrast no matrix inversion is required for the solution of the state estimates when using the suboptimal design. The motivation for investigating suboptimal filter designs was precisely this need to minimize the number of computations required for the solution of the state estimates.

The penalty incurred by implementing the 3 state filter over the 9 state filter is investigated in this section. As seen in Section 3, the 9 state filter is parameterized by the same quantities as the 3 state filter. The parameters of interest are the quantities a_1, a_2, a_3 whose optimal values are $a_1 = a_2 = a_3 = .1$. In implementing this filter, the state dependent R matrix is used. Results of the simulation for all 12 flight passes are listed in Table 38. Comparison of these results with those of Table 33 for the 3 state filter indicates that, on the average, the 9 state filter results in a 2.7% decrease in position RMS error, a 10.9% decrease in the velocity RMS error, and a 16.4% decrease in the acceleration RMS error. These results are encouraging because they indicate that the improvement in state estimates is marginal. It is therefore reasonable to assert that the ease of implementation of the 3 state filter outweighs the marginal gain in accuracy incurred with use of the 9 state design. The conclusion, then, is that the 3 state filter derived in this report is most suitable for implementation in a fire control system.

TABLE 38

TOTAL RMS ERRORS FOR 9 STATE FILTER WITH R , STATE DEPENDENT

Flight Pass	RMS ϵ_{pos}	RMS ϵ_{vel}	RMS ϵ_{acc}
1	6.6	12.2	13.4
2	7.1	13.2	14.2
3	7.1	12.9	14.7
4	6.6	11.8	13.4
5	7.0	12.6	13.5

TABLE 38 (Cont)

<u>Flight Pass</u>	<u>RMS ϵ_{pos}</u>	<u>RMS ϵ_{vel}</u>	<u>RMS ϵ_{acc}</u>
6	6.7	13.0	16.8
7	6.9	13.0	11.4
8	6.8	13.0	16.6
9	7.3	14.9	19.5
10	7.3	11.6	19.2
11	7.6	15.4	19.6
12	8.0	13.9	11.5
Average	7.1	13.1	15.3

APPENDIX A

THE KALMAN FILTER EQUATIONS

The discrete form of the Kalman filter equations are listed here for reference together with the names commonly associated with some of the terms.

$$\hat{X}_{n/n} = F_{n-1} \hat{X}_{n-1/n-1} + K_n [Z_n - H_n F_{n-1} \hat{X}_{n-1/n-1}]$$

$$K_n = P_n H_n^T [H_n P_n H_n^T + R_n]^{-1}$$

$$P_{n+1} = F_n [I - K_n H_n] P_n [I - K_n H_n]^T F_n^T + G_n Q_n G_n^T \\ + F_n K_n R_n K_n^T F_n^T$$

$\hat{X}_{n/n}$ = estimate of state

F_{n-1} = state transition matrix

K_n = gain

Z_n = observation matrix

P_n = state estimation error covariance matrix

Q_n = plant covariance matrix

R_n = observation covariance matrix

$[Z_n - H_n F_{n-1} \hat{X}_{n-1/n-1}]$ = innovation

The discrete form of the state equations arise from the discretization of the continuous first order differential equations called the plant model.

$$X_{n+1} = F_n X_n + G_n \omega_n$$

The sensor model is linear in the state variable.

$$Z_n = H_n X_n + D_n v_n$$

The underlying statistical assumptions of the Kalman theory are that the additive noise in the plant and observation models is Gaussian. The restriction on the observation model is that it be linear in the state variables. If the condition of Gaussian statistics is violated, then the Kalman filter is still the best linear state estimator.

APPENDIX B

DISCRETE PLANT EQUATIONS

Here the discrete form of the plant equations is derived. The identification of the terms in the resulting equations which are needed for implementation of the Kalman filter equations is made.

In continuous form, the plant model is written as

$$(B-1) \quad \dot{\underline{X}} = \underline{F} \underline{X} + \underline{G} \underline{u}$$

Let Φ satisfy the differential equation

$$\dot{\Phi} = F\Phi \text{ with } \Phi(0) = I.$$

Then,

$$\frac{d}{dt} (\Phi^{-1} X) = \dot{\Phi}^{-1} X + \Phi^{-1} \dot{X}$$

But,

$$(B-2) \quad \Phi \Phi^{-1} = I$$

Differentiating both sides, obtain

$$\dot{\Phi}^{-1} = -\Phi^{-1} \dot{\Phi} \Phi^{-1}$$

Thus,

$$\begin{aligned} \frac{d}{dt} (\Phi^{-1} X) &= -\Phi^{-1} \dot{\Phi} \Phi^{-1} X + \Phi^{-1} \dot{X} \\ &= \Phi^{-1} (\dot{X} - FX) = \Phi^{-1} G u \end{aligned}$$

$$\int_0^t d(\Phi^{-1} X) = \int_0^t \Phi^{-1} G u \, d\tau$$

Or,

$$\Phi^{-1}(t) X(t) = \Phi^{-1}(0) X(0) + \int_0^t \Phi^{-1}(\tau) G u \, d\tau$$

$$X(t) = \Phi(t) \Phi^{-1}(0) X(0) + \Phi(t) \int_0^t \Phi^{-1}(\tau) G u \, d\tau$$

AD-A036 440

FRANKFORD ARSENAL PHILADELPHIA PA

F/G 19/5

INVESTIGATION OF KALMAN FILTER DESIGNS FOR AIR DEFENSE BASED ON--ETC(1)

SEP 76 W J DZIWA, L VITALE, M MINTZ

UNCLASSIFIED

FA-TR-76058

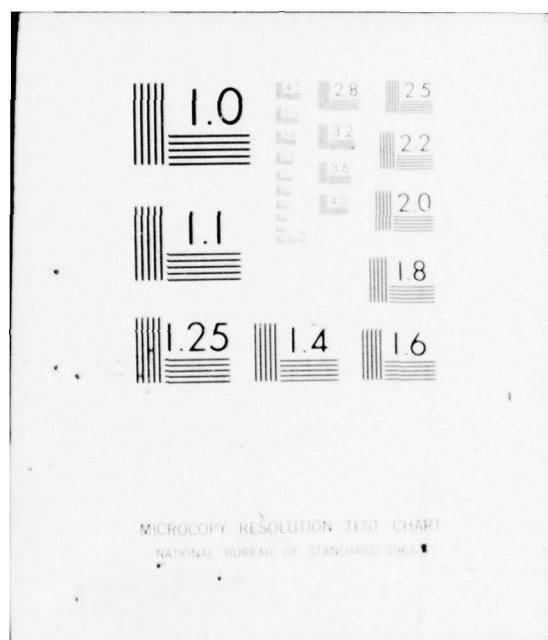
NL

2 OF 2
AD
A036440



END

DATE
FILMED
8-77



Since $\phi^{-1}(\tau) = \phi(-\tau)$

$$(B-3) \quad X(t) = \phi(t) X(0) + \int_0^t \phi(t-\tau) G(\tau) \omega(\tau) d\tau$$

This equation is discretized by replacing t with $(n+1)\Delta$ where Δ is the time between samples and $n = 0, 1, 2, \dots$. Thus,

$$X[(n+1)\Delta] = \phi[(n+1)\Delta] X(0) + \phi[(n+1)\Delta] \int_0^{(n+1)\Delta} \phi(-\tau) G\omega(\tau) d\tau$$

But, $\phi(t_1 + t_2) = \phi(t_1) \phi(t_2)$. Hence,

$$\begin{aligned} X[(n+1)\Delta] &= \phi(\Delta) [\phi(n\Delta) X(0) + \phi(n\Delta) \int_0^{n\Delta} \phi(-\tau) G\omega(\tau) d\tau \\ &\quad + \phi(n\Delta) \int_{n\Delta}^{(n+1)\Delta} \phi(-\tau) G\omega(\tau) d\tau] \\ &= \phi(\Delta) X(n\Delta) + \phi(\Delta) \phi(n\Delta) \int_{n\Delta}^{(n+1)\Delta} \phi(-\tau) G\omega(\tau) d\tau \end{aligned}$$

$$(B-4) \quad X[(n+1)\Delta] = \phi(\Delta) X(n\Delta) + \phi(\Delta) \int_{n\Delta}^{(n+1)\Delta} \phi(n\Delta - \tau) G\omega(\tau) d\tau$$

Now replace $\omega(\tau)$ by $\tilde{\omega}$, where $\tilde{\omega}$ is the time averaged white noise. That is,

$$(B-5) \quad \tilde{\omega} = \frac{1}{\Delta} \int_{n\Delta}^{(n+1)\Delta} \omega(t) dt$$

Also, change variables of integration by letting $-\tau \rightarrow n\Delta - \tau$. Then,

$$\begin{aligned} X[(n+1)\Delta] &= \phi(\Delta) X(n\Delta) + \phi(\Delta) \int_0^{\Delta} \phi(-\tau) G\tilde{\omega} d\tau \\ &= \phi(\Delta) X(n) + \int_0^{\Delta} \phi(\tau) G\tilde{\omega} d\tau \end{aligned}$$

Changing notation so that $X_n = X(n)$, the discrete form of the plant model is

$$(B-6) \quad X_{n+1} = \phi(\Delta) X_n + \int_0^{\Delta} \phi(\tau) G\tilde{\omega} d\tau$$

From $X_{n+1} = F_n X_n + G_n \omega_n$, we identify

$$(B-7) \quad F_n \rightarrow \Phi(\Delta)$$

and

$$(B-8) \quad G_n \rightarrow \int_0^\Delta \Phi(\tau) G d\tau$$

From Equation (B-3), the solution to the homogeneous part of the state equation is

$$(B-9) \quad X(t) = \Phi(t) X(0)$$

Thus, $\Phi(t)$ can be obtained explicitly from a solution to the homogeneous state equation. Consider the plant model

$$\begin{aligned} \dot{X}_1 &= X_2 \\ \dot{X}_2 &= X_3 \\ \dot{X} &= -aX_3 + b\omega \end{aligned}$$

The homogeneous equations are obtained by setting $b = 0$. These are then solved to give

$$X_1(t) = X_1(0) + X_2(0)t + \left[\frac{1}{a}t + \frac{1}{a^2}(e^{-at} - 1) \right] X_3(0)$$

$$X_2(t) = X_2(0) + \frac{1}{a}(1 - e^{-at}) X_3(0)$$

$$X_3(t) = e^{-at} X_3(0)$$

By inspection,

$$(B-10) \quad \Phi(\Delta) = \begin{bmatrix} 1 & \Delta & \frac{1}{a}\Delta + \frac{1}{a^2}(e^{-a\Delta} - 1) \\ 0 & 1 & \frac{1}{a}(1 - e^{-a\Delta}) \\ 0 & 0 & e^{-a\Delta} \end{bmatrix}$$

APPENDIX C

ACCELERATION CORRELATION FUNCTIONS

The correlation function for each of the acceleration processes described in this report is derived below.

Consider the simplest acceleration model where the acceleration is described as a first order Markov process,

$$(C-1) \quad \dot{X}_3 = -aX_3 + bu$$

Assuming the initial condition to be zero, take the Laplace transform of both sides to obtain

$$(C-2) \quad S X_3 (S) = -aX_3 (S) + bu (S)$$

The transfer function $H (S)$ is

$$H(S) \equiv \frac{X_3 (S)}{u(S)} = \frac{b}{S+a}$$

$$(C-3) \quad |H (j\omega)|^2 = \frac{b^2}{\omega^2 + a^2}$$

If S_{x3} is the power density spectrum of the acceleration and S_u the power density spectrum of the white noise, then

$$(C-4) \quad S_{x3} (\omega) = |H (j\omega)|^2 S_u (u)$$

Since the power density spectrum of white noise is unity,

$$(C-5) \quad S_{x3} (\omega) = |H (j\omega)|^2 = \frac{b^2}{\omega^2 + a^2}$$

The correlation function $R_{x3} (\tau)$ is the inverse Fourier transform of the power density. Thus,

$$\begin{aligned} (C-6) \quad R_{x3} (\tau) &= \frac{1}{2\pi} \int_{-\infty}^{\infty} S_{x3} (\omega) e^{i\omega\tau} d\omega \\ &= \frac{1}{2\pi} \int_{-\infty}^{\infty} \frac{b^2}{\omega^2 + a^2} e^{i\omega\tau} d\omega \\ &= \frac{b^2}{2a} e^{-a|\tau|} \end{aligned}$$

The derivation of the correlation function for the acceleration, modeled as a second order Markov process proceeds along similar lines.

The process is described by

$$(C-7) \quad \dot{X}_3 = X_4 + bu$$

$$(C-8) \quad \dot{X}_4 = -\alpha^2 X_3 - 2\beta X_4 + (\alpha - 2\beta) bu$$

$$\text{with } \alpha^2 = \omega_0^2 + \beta^2$$

The Laplace transform of each equation gives

$$(C-9) \quad S X_3(S) = X_4(S) + bu(S)$$

$$(C-10) \quad S X_4(S) = -\alpha^2 X_3(S) - 2\beta X_4(S) + (\alpha - 2\beta) bu(S)$$

From these, one obtains the transfer function

$$\begin{aligned} (C-11) \quad H(S) &= \frac{X_3(S)}{u(S)} = \frac{b(S+\alpha)}{S^2 + 2\beta S + \alpha^2} \\ &= \frac{b(S+\alpha)}{(S+2\beta S + \omega_0^2 + \beta^2)} \\ &= \frac{b(S+\alpha)}{(S+\beta+j\omega_0)(S+\beta-j\omega_0)} \end{aligned}$$

Define,

$$G(S) \equiv H(S) H(-S) = \frac{-b^2 (S+\alpha)}{(S+\beta+j\omega_0)(S+\beta-j\omega_0)} \cdot \frac{(S-\alpha)}{(S-\beta-j\omega_0)(S-\beta+j\omega_0)}$$

Let

$$p = \beta - j\omega_0$$

Then,

$$(C-12) \quad G(S) = \frac{-b^2 (S+\alpha)(S-\alpha)}{(S+p^*)(S+p)(S-p)(S-p^*)}$$

$$\begin{aligned} (C-13) \quad R(\tau) &= \frac{1}{2\pi} \int_{-\infty}^{\infty} G(j\omega) e^{j\omega\tau} d\omega \\ &= \frac{-i}{2\pi} \int G(S) e^{S\tau} dS \\ &= \frac{ib^2}{2\pi} \int_C \frac{(S+\alpha)(S-\alpha) e^{S\tau} dS}{(S+p^*)(S+p)(S-p^*)(S-p)} \end{aligned}$$

where the contour of integration is shown in Figure C-1.

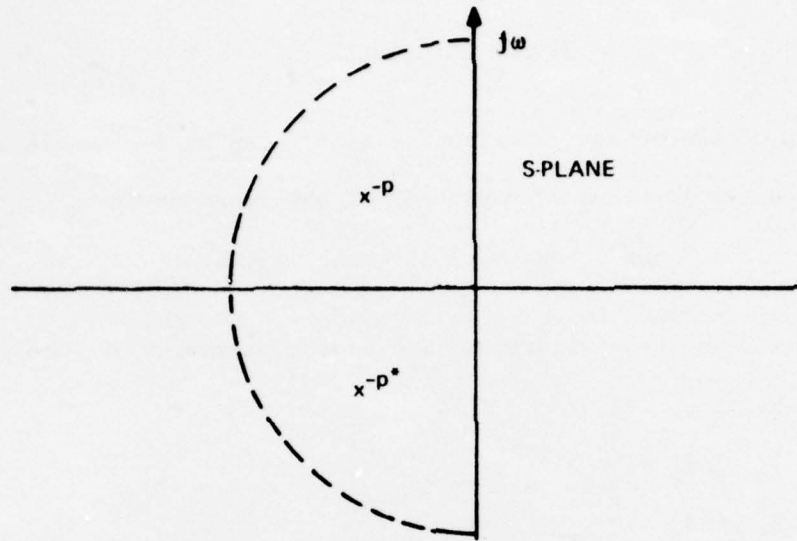


Figure C-1. Contour of Integration

This contour encloses poles at $S = -p$ and $S = -p^*$. From residue theory,

$$R_{x_3}(\tau) = -b^2 (R_1 + R_2)$$

where R_1 is the residue of pole $-p^*$ and R_2 is the residue of pole $-p$. Thus,

$$R_1 = \frac{(S+\alpha)}{(S+p)} \frac{(S-\alpha)}{(S-p^*)} \frac{e^{S\tau}}{(S-p)} \Big|_{S = -p^*}$$

$$= -\frac{1}{4\beta} e^{-p^*\tau}$$

$$R_2 = \frac{(S+\alpha)}{(S+p^*)} \frac{(S-\alpha)}{(S-p^*)} \frac{e^{S\tau}}{(S-p)} \Big|_{S = -p}$$

$$R_2 = R_1^*$$

Thus,

$$R(\tau) = -b^2 (R_1 + R_1^*) = -2 b^2 \operatorname{Re} (R_1)$$

Or,

$$(C-14) R(\tau) = \frac{b^2}{2\beta} e^{-\beta\tau} \cos(\omega_0\tau)$$

APPENDIX D

SENSOR COVARIANCE

The elements of the sensor covariance matrix \underline{R} can be derived in a straightforward manner in terms of variances of the range error σ_R^2 , azimuth and elevation errors σ_φ^2 and σ_θ^2 as well as their derivatives $\sigma_{\dot{R}}^2$, $\sigma_{\dot{\varphi}}^2$, $\sigma_{\dot{\theta}}^2$.

These quantities are assumed to be known in advance. The geometry of Figure 1 of Section 3 is used in the derivation. The sensor variables of interest are:

$$(D-1) \quad Z_1 = R_s \sin\theta_s \cos\varphi_s$$

$$(D-2) \quad Z_2 = R_s \sin\theta_s \sin\varphi_s$$

$$(D-3) \quad Z_3 = R_s \cos\theta_s$$

$$(D-4) \quad Z_4 = \dot{R}_s \sin\theta_s \cos\varphi_s + R_s \cos\theta_s \cos\varphi_s \dot{\theta}_s - R_s \sin\theta_s \sin\varphi_s \dot{\varphi}_s$$

$$(D-5) \quad Z_5 = \dot{R}_s \sin\theta_s \sin\varphi_s + R_s \cos\theta_s \sin\varphi_s \dot{\theta}_s + R_s \sin\theta_s \cos\varphi_s \dot{\varphi}_s$$

$$(D-6) \quad Z_6 = \dot{R}_s \cos\theta_s - R_s \sin\theta_s \dot{\theta}_s$$

where the subscript s is used to indicate that the associated quantity is the sensed or measured value. Since the measured quantities are contaminated with noise, they can be written as:

$$R_s = R + \Delta R$$

$$\dot{R}_s = \dot{R} + \Delta \dot{R}$$

$$\theta_s = \theta + \Delta\theta$$

$$\dot{\theta}_s = \dot{\theta} + \Delta \dot{\theta}$$

$$\varphi_s = \varphi + \Delta\varphi$$

$$\dot{\varphi}_s = \dot{\varphi} + \Delta \dot{\varphi}$$

where R , θ , φ , \dot{R} , $\dot{\theta}$, $\dot{\varphi}$ are the noise-free sensor variables. The assumption on the noise terms ΔR , $\Delta\theta$, $\Delta\varphi$, $\Delta \dot{R}$, $\Delta \dot{\theta}$, $\Delta \dot{\varphi}$, is that they are stationary and independent with zero mean values. That is, $E(\Delta R) = 0$, $E(\Delta\theta) = 0$, $E(\Delta\varphi) = 0$, $E(\Delta \dot{R}) = 0$, $E(\Delta \dot{\theta}) = 0$, $E(\Delta \dot{\varphi}) = 0$; $E(\Delta R \Delta\theta) = E(\Delta R) E(\Delta\theta)$ etc. from which it follows that $E(R_s) = R$, $E(\theta_s) = \theta$, etc.

If X_1 , X_2 , X_3 are the variables corresponding to the coordinates of target position along X , Y , and Z respectively, and X_4 , X_5 , X_6 are the corresponding velocity terms, then the quantities entering in the sensor covariance matrix are:

$$(D-7) \quad \sigma_{x_i}^2 = E(Z_i - X_i)^2 \quad (i=1, \dots, 6)$$

and

$$(D-8) \quad \sigma_{x_i x_j} = E[(Z_i - X_i)(Z_j - X_j)]$$

Thus, for example, $\sigma_{x_i x_j}$ is the R_{ij} term of Section 3 for $i = 1, 2, 3$.

Consider the quantity $\sigma_{x_1}^2$. Using definition (D-7),

$$\begin{aligned}\sigma_{x_1}^2 &= E (Z_1 - X_1)^2 = E (Z_1^2) - X_1^2 \\ &= E \{[(R + \Delta R) \sin (\theta + \Delta \theta) \cos (\varphi + \Delta \varphi)]^2\} - X_1^2 \\ &= (R^2 + \sigma_R^2) \sigma_\varphi^2 \sin^2 \varphi \sin^2 \theta + (R^2 + \sigma_\theta^2) \sigma_\theta^2 \cos^2 \theta \cos^2 \varphi \\ &\quad + (R^2 + \sigma_R^2) \sigma_\theta^2 \sigma_\varphi^2 \cos^2 \theta \sin^2 \varphi + \sigma_R^2 \sin^2 \theta \cos^2 \varphi\end{aligned}$$

where we have used the result $E (\Delta R)^2 = \sigma_R^2$, $E (\Delta \theta)^2 = \sigma_\theta^2$, $E (\Delta \varphi)^2 = \sigma_\varphi^2$ and made the small angle approximations $\sin \Delta \theta \approx \Delta \theta$, $\sin \Delta \varphi \approx \Delta \varphi$, $\cos \Delta \theta \approx 1$, $\cos \Delta \varphi \approx 1$. Assuming that $R^2 \gg \sigma_R^2$, the expression for $\sigma_{x_1}^2$, reduces to

$$\begin{aligned}\text{(D-9)} \quad \sigma_{x_1}^2 &= R^2 [\sigma_\varphi^2 \sin^2 \theta \sin^2 \varphi + \sigma_\theta^2 \sigma_\varphi^2 \cos^2 \theta \sin^2 \varphi + \sigma_\theta^2 \cos^2 \theta \cos^2 \varphi] \\ &\quad + \sigma_R^2 \sin^2 \theta \cos^2 \varphi.\end{aligned}$$

The expression for $\sigma_{x_2}^2$ is obtained by simply replacing φ by $90^\circ - \varphi$ in Equation (D-9). Thus,

$$\begin{aligned}\text{(D-10)} \quad \sigma_{x_2}^2 &= R^2 [\sigma_\varphi^2 \sin^2 \theta \cos^2 \varphi + \sigma_\theta^2 \sigma_\varphi^2 \cos^2 \theta \cos^2 \varphi + \sigma_\theta^2 \cos^2 \theta \sin^2 \varphi] \\ &\quad + \sigma_R^2 \sin^2 \theta \sin^2 \varphi.\end{aligned}$$

Replacing θ by $90^\circ - \theta$ and φ by 0° in Equation (D-9), we obtain $\sigma_{x_3}^2$. Thus,

$$\text{(D-11)} \quad \sigma_{x_3}^2 = R^2 \sigma_\theta^2 \sin^2 \theta + \sigma_R^2 \cos^2 \theta$$

The quantity $\sigma_{x_1 x_2}$ is obtained in a similar manner, although the algebra is a bit more tedious.

$$\sigma_{x_1 x_2} = E[(Z_1 - X_1)(Z_2 - X_2)] = E(Z_1 Z_2) - X_1 X_2$$

$$= E\{[(R+\Delta R) \sin(\theta+\Delta\theta) \cos(\varphi+\Delta\varphi) [(R+\Delta R) \sin(\theta+\Delta\theta) \sin(\varphi+\Delta\varphi)]\} - X_1 X_2$$

$$= -(R^2 + \sigma_R^2) \sigma_\theta^2 \cos^2 \theta \sin \varphi \cos \varphi + (R^2 + \sigma_R^2) \sigma_\varphi^2 \sin^2 \theta \cos \varphi \sin \varphi$$

$$+ (R^2 + \sigma_R^2) \sigma_\varphi^2 \sigma_\theta^2 \cos^2 \theta \sin \varphi \cos \varphi - \sigma_R^2 \sin^2 \theta \sin \varphi \cos \varphi$$

Again, assuming that $R^2 \gg \sigma_R^2$, $\sigma_{x_1 x_2}$ reduces to

$$(D-12) \sigma_{x_1 x_2} = R^2 \cos^2 \varphi \sin^2 \varphi [\sigma_\theta^2 \cos^2 \theta - \sigma_\theta^2 \sin^2 \theta] - \sigma_R^2 \sin^2 \theta \cos \varphi \sin \varphi$$

where the additional assumption that $\sigma_\theta^2 \cos^2 \theta \gg \sigma_\theta^2 \sin^2 \theta$ was used.

The derivation of the quantities $\sigma_{x_1 x_3}$ and $\sigma_{x_2 x_3}$ proceeds in exactly the same manner. The results are

$$(D-13) \sigma_{x_1 x_3} = (\sigma_R^2 - R^2 \sigma_\theta^2) \sin \theta \cos \theta \cos \varphi$$

$$(D-14) \sigma_{x_2 x_3} = (\sigma_R^2 - R^2 \sigma_\theta^2) \cos \theta \sin \theta \sin \varphi$$

For the velocity term Z_4 , σ_{x_4} is computed as follows:

$$\sigma_{x_4}^2 = E(Z_4 - X_4)^2 = E(Z_4^2) - X_4^2$$

$$(D-15) \sigma_{x_4}^2 = (R^2 + \sigma_R^2) [\sigma_\varphi^2 \sin^2 \theta \sin^2 \varphi + \sigma_\theta^2 \cos^2 \theta \cos^2 \varphi$$

$$+ \sigma_\theta^2 \sigma_\varphi^2 \cos^2 \theta \sin^2 \varphi] + \sigma_R^2 \sin^2 \theta \cos^2 \varphi$$

$$+ R^2 (\dot{\theta}^2 + \sigma_\theta^2) [\sigma_\varphi^2 \cos^2 \theta \sin^2 \varphi + \sigma_\theta^2 \sin^2 \theta \cos^2 \varphi$$

$$+ \sigma_\theta^2 \sigma_\varphi^2 \sin^2 \theta \sin^2 \varphi] + R^2 \sigma_\theta^2 \cos^2 \theta \cos^2 \varphi$$

$$+ R^2 (\dot{\varphi}^2 + \sigma_\varphi^2) [\sigma_\varphi^2 \sin^2 \theta \cos^2 \varphi + \sigma_\theta^2 \cos^2 \theta \sin^2 \varphi$$

$$\begin{aligned}
& + \sigma_{\theta}^2 \sigma_{\varphi}^2 \cos^2 \theta \cos^2 \varphi] + R^2 \sigma_{\dot{\varphi}}^2 \sin^2 \theta \sin^2 \varphi \\
& + 2\ddot{R}R (\dot{\varphi} \sigma_{\varphi}^2 \sin^2 \theta \cos \varphi + \dot{\theta} \sigma_{\varphi}^2 \sin \theta \cos \theta \sin \varphi \\
& - \sigma_{\theta}^2 \dot{\varphi} \cos \theta \cos \varphi) \sin \varphi
\end{aligned}$$

where again the assumptions $R^2 \gg \sigma_R^2$ and $\sigma_{\theta}^2 \ll 1$, $\sigma_{\varphi}^2 \ll 1$ were used. To obtain $\sigma_{x_5}^2$ simply replace φ by $90^\circ - \varphi$. The result is,

$$\begin{aligned}
(D-16) \sigma_{x_5}^2 = & (R^2 + \sigma_R^2) [\sigma_{\varphi}^2 \sin^2 \theta \cos^2 \varphi + \sigma_{\theta}^2 \cos^2 \theta \sin^2 \varphi \\
& + \sigma_{\theta}^2 \sigma_{\varphi}^2 \cos^2 \theta \cos^2 \varphi] + \sigma_R^2 \sin^2 \theta \sin^2 \varphi + R^2 (\dot{\theta}^2 + \sigma_{\dot{\theta}}^2) \\
& [\sigma_{\varphi}^2 \cos^2 \theta \cos^2 \varphi + \sigma_{\theta}^2 \sin^2 \theta \sin^2 \varphi + \sigma_{\theta}^2 \sigma_{\varphi}^2 \sin^2 \theta \cos^2 \varphi] \\
& + R^2 \sigma_{\dot{\theta}}^2 \cos^2 \theta \sin^2 \varphi + R^2 (\dot{\varphi}^2 + \sigma_{\dot{\varphi}}^2) [\sigma_{\varphi}^2 \sin^2 \theta \sin^2 \varphi \\
& + \sigma_{\theta}^2 \cos^2 \theta \cos^2 \varphi + \sigma_{\theta}^2 \sigma_{\varphi}^2 \cos^2 \theta \cos^2 \varphi] + R^2 \sigma_{\dot{\varphi}}^2 \sin^2 \theta \cos^2 \varphi \\
& + 2\ddot{R}R (\dot{\varphi} \sigma_{\varphi}^2 \sin^2 \theta \sin \varphi + \dot{\theta} \sigma_{\varphi}^2 \sin \theta \cos \theta \sin \varphi - \sigma_{\theta}^2 \dot{\varphi} \cos^2 \theta \sin \varphi) \cos \varphi
\end{aligned}$$

The derivation for $\sigma_{x_6}^2$ results in

$$\begin{aligned}
(D-17) \sigma_{x_6}^2 = & R^2 \sigma_{\dot{\theta}}^2 \sin^2 \theta + R^2 (\dot{\theta}^2 + \sigma_{\dot{\theta}}^2) \sigma_{\theta}^2 \cos^2 \theta \\
& + \sigma_R^2 \dot{\theta}^2 \sin^2 \theta + (R^2 + \sigma_R^2) \sigma_{\theta}^2 \sin^2 \theta \\
& + \sigma_R^2 \cos^2 \theta + 2\ddot{R}R \sigma_{\theta}^2 \sin \theta \cos \theta
\end{aligned}$$

Now compute the terms which couple the position-velocity measurements. For $\sigma_{x_1 x_4}$ obtain

$$\begin{aligned}
\sigma_{x_1 x_4} & = E[(Z_1 - X_1)(Z_4 - X_4)] = E(Z_1 X_4) - X_1 X_4 \\
(D-18) \sigma_{x_1 x_4} & = \ddot{R}R (\sigma_{\varphi}^2 \sin^2 \theta \sin^2 \varphi + \sigma_{\theta}^2 \cos^2 \theta \cos^2 \varphi
\end{aligned}$$

$$\begin{aligned}
& + \sigma_{\theta}^2 \sigma_{\varphi}^2 \cos^2 \theta \sin^2 \varphi) - \dot{\theta} \sigma_{\varphi}^2 R^2 \sin \theta \cos \theta \sin^2 \varphi \\
& - \dot{\varphi} \sigma_{\theta}^2 R^2 \cos^2 \theta \sin \varphi \cos \varphi
\end{aligned}$$

whereas $\sigma_{x_2 x_5}$ is obtained by replacing φ by $90^\circ - \varphi$ in the expression for $\sigma_{x_1 x_4}$. Thus,

$$\begin{aligned}
\text{(D-19) } \sigma_{x_2 x_5} &= RR (\sigma_{\varphi}^2 \sin^2 \theta \cos^2 \varphi + \sigma_{\theta}^2 \cos^2 \theta \sin^2 \varphi \\
&+ \sigma_{\theta}^2 \sigma_{\varphi}^2 \cos^2 \theta \cos^2 \varphi) - \dot{\theta} \sigma_{\varphi}^2 R^2 \sin \theta \cos \theta \cos^2 \varphi \\
&- \dot{\varphi} \sigma_{\theta}^2 R^2 \cos^2 \theta \cos \varphi \sin \varphi
\end{aligned}$$

The term $\sigma_{x_3 x_6}$ is simply

$$\text{(D-20) } \sigma_{x_3 x_6} = RR \sigma_{\theta}^2 \sin^2 \theta$$

The remaining covariance terms are $\sigma_{x_1 x_5}$, $\sigma_{x_1 x_6}$, $\sigma_{x_2 x_4}$, $\sigma_{x_2 x_6}$, $\sigma_{x_3 x_4}$ and $\sigma_{x_3 x_5}$. However, they are not used in the report and will therefore not be derived here.

APPROXIMATIONS TO THE R MATRIX

Successive implementation of the R matrix using the above expressions at each iteration time can be a time-consuming process. It is therefore desirable to approximate these expressions, ideally by some constant numerical values. This is necessary, especially if one attempts to achieve a steady state filter design.

The procedure for simplifying the expressions will be simply the following: Each expression will be averaged over the sensor variables. We begin with $\sigma_{x_1}^2$. The only variables appearing in $\sigma_{x_1}^2$ are R , θ and φ . Average φ from 0 to 2π and θ from 0 to $\pi/2$ and assume that $\sigma_{\theta}^2 \sigma_{\varphi}^2 \ll \sigma_{\phi}^2$. Then,

$$\frac{1}{(2\pi)(\pi/2)} \int_0^{\pi/2} \int_0^{2\pi} \sigma_{x_1}^2 d\theta d\varphi = 1/4 \sigma_R^2 + 1/4 R^2 (\sigma_{\theta}^2 + \sigma_{\varphi}^2)$$

Similarly, for $\sigma_{x_2}^2$ and $\sigma_{x_3}^2$,

$$\frac{1}{(2\pi)(\pi/2)} \int_0^{\pi/2} \int_0^{2\pi} \sigma_{x_2}^2 d\theta d\varphi = 1/4 \sigma_R^2 + 1/4 R^2 (\sigma_\theta^2 + \sigma_\varphi^2)$$

$$\frac{1}{(2\pi)(\frac{\pi}{2})} \int_0^{\pi/2} \int_0^{2\pi} \sigma_{x_3}^2 d\theta d\varphi = 1/2 \sigma_R^2 + 1/2 R^2 \sigma_\theta^2$$

The resulting expressions, which are functions of R, are now averaged over R where the limits on R are 0 and 6×10^3 m (a rather extreme upper limit).

The quantities σ_R^2 , σ_θ^2 , σ_φ^2 are taken to be 100 m^2 , 1 m^2 , and 1 m^2 respectively.

This results in the numerical values

$$(D-21) \sigma_{x_1}^2 = \sigma_{x_2}^2 = 31$$

$$(D-22) \sigma_{x_3}^2 = 56$$

In a similar manner, one obtains, for $\sigma_{x_4}^2$, $\sigma_{x_5}^2$, and $\sigma_{x_6}^2$

$$(D-23) \sigma_{x_4}^2 = \sigma_{x_5}^2 = 12.5$$

$$(D-24) \sigma_{x_6}^2 = 16$$

where the limits on \dot{R} , $\dot{\theta}$ and $\dot{\varphi}$ are, respectively, $\pm 300 \text{ m/sec}$, $\pm 1 \text{ rad/sec}$, and $\pm 1 \text{ rad/sec}$ with $\sigma_{\dot{R}} = 5 \text{ m/sec}$ and $\sigma_{\dot{\theta}} = \sigma_{\dot{\varphi}} = .5 \text{ rad/sec}$.

APPENDIX E

INITIALIZATION OF POSITION-VELOCITY, VELOCITY-VELOCITY TERMS OF P_0

The velocity variance as well as the position-velocity variance terms are computed here for initialization of the P matrix. The assumption used is that the sensor provides position data only. Consider the $P_{12}(0)$ terms which couples the x-position with the x-velocity. Since only position data is available, the estimate of the initial x-component of velocity, $X_2(\Delta)$, is using the notation of Section 4,

$$(E-1) \quad X_2(\Delta) = \frac{Z_1(\Delta) - Z_1(0)}{\Delta}$$

where Δ is the time between measurements $Z_1(0)$ and $Z_1(\Delta)$. Thus,

$$\begin{aligned} (E-2) \quad P_{12}(0) &= E(\hat{X}_1(\Delta) - X_1(\Delta))(\hat{X}_2(\Delta) - X_2(\Delta)) \\ &= \frac{1}{\Delta} E(Z_1(\Delta) - X_1(\Delta))[(Z_1(\Delta) - X_1(\Delta)) - (Z_1(0) - X_1(0))] \\ &= \frac{1}{\Delta} \sigma_{x_1}^2(\Delta) = \frac{1}{\Delta} R_{11}(0) \end{aligned}$$

where the position measurements are assumed to be uncorrelated in time. Similarly, $P_{15}(0)$ which couples the x-coordinate of target position with the y-component of velocity is computed by writing

$$\begin{aligned} (E-3) \quad P_{15}(0) &= E(\hat{X}_1(\Delta) - X_1(\Delta))(\hat{X}_5(\Delta) - X_5(\Delta)) \\ &= \frac{1}{\Delta} E(Z_1(\Delta) - X_1(\Delta))[(Z_2(\Delta) - X_5(\Delta)) - (Z_2(0) \\ &\quad - X_5(0))] \\ &= \frac{1}{\Delta} \sigma_{x_1 x_5}^2 = \frac{1}{\Delta} R_{12} \end{aligned}$$

Similarly, the remaining terms may be written down by inspection. In summary,

$$(E-4) \quad P_{12}(0) = \frac{1}{\Delta} R_{11}(0)$$

$$(E-5) \quad P_{15}(0) = \frac{1}{\Delta} R_{12}(0)$$

$$(E-6) \quad P_{18}(0) = \frac{1}{\Delta} R_{13}(0)$$

$$(E-7) \quad P_{27}(0) = P_{72}(0) = \frac{1}{\Delta} R_{13}(0)$$

$$(E-8) \quad P_{75}(0) = P_{57}(0) = \frac{1}{\Delta} R_{23}(0)$$

$$(E-9) \quad P_{78}(0) = P_{87}(0) = \frac{1}{\Delta} R_{33}(0)$$

$$(E-10) \quad P_{14}(0) = P_{41}(0) = \frac{1}{\Delta} R_{12}(0)$$

$$(E-11) \quad P_{45}(0) = P_{54}(0) = \frac{1}{\Delta} R_{22}(0)$$

$$(E-12) \quad P_{48}(0) = P_{84}(0) = \frac{1}{\Delta} R_{23}(0)$$

where Δ is the time interval between the first and last position measurements prior to filtering.

APPENDIX F

COMPUTER SOFTWARE

This appendix provides the computer software and documentation for the nine state Kalman filter as described in Section 3. The data presented in this report was generated by several computer programs. The most general program is presented in this Appendix. It was constructed in a modular form to permit experimentation with a variety of filtering and sensing algorithms. The user may, for example, modify the program's form to obtain: a steady state filter; a position rate sensor; a correlated noise sensor or a filtering arrangement composed of three independent three state filters.

The computer language implemented on the CDC 6500 computer was Extended Fortran IV. All of the Fortran statements used should be compatible with similar systems. The only exception, whose form might vary for a different compiler, is the non-standard ENTRY into a subroutine. The executable field length necessary to "run" in program is 52000 octal words and the corresponding system time is 225.5 system seconds.

The format for the documentation is:

F-1) A narrative description of the subroutines used to simulate and solve the estimation problem.

F-2) An analytical description of the main program and the subroutines containing: flow charts, routine summary, and an abridged variable list.

F-3) A detailed description of the subroutines and corresponding changes necessary to obtain desired filtering and sensing algorithms.

F-4) An index to subroutines including alphabetic letter reference appearing on flow chart diagrams, memory requirements and calling and returning arguments.

F-5) A computer listing of the main program and corresponding subroutines.

STATE VECTOR

	<u>Appears in Report</u>	<u>Appears in Appendix</u>
X(1)	position in X-direction	position in X-direction
X(2)	velocity in X-direction	position in Y-direction
X(3)	acceleration in X-direction	position in Z-direction
X(4)	position in Y-direction	velocity in X-direction
X(5)	velocity in Y-direction	velocity in Y-direction
X(6)	acceleration in Y-direction	velocity in Z-direction
X(7)	position in Z-direction	acceleration in X-direction
X(8)	velocity in Z-direction	acceleration in Y-direction
X(9)	acceleration in Z-direction	acceleration in Z-direction

NOTE:

(1) The notation in the main report and in this Appendix differ in the description of the target state vector.

This difference in ordering of the state variables affects the arrangement of the elements in the P, R, K, F, G, GH and XH matrices.

(2) All data used and computed in the main program and corresponding subroutines are expressed in the M-K-S units. Other units can be used, if desired, by changing the appropriate numerical specification of the target state vector and sensor accuracy.

(3) The abbreviations used throughout the appendix are:

TA = target aircraft
TAFP = target aircraft flight path - refers to the state along the flight path
MS = mean square
MSE = mean square error

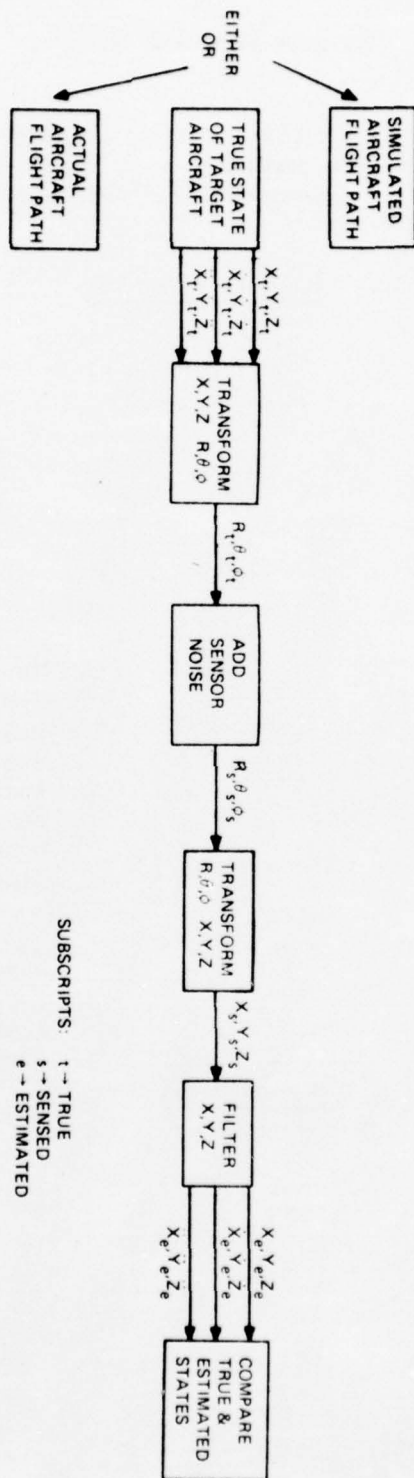
(4) The solution in the state estimator requires the inversion of a 3×3 matrix. The subroutine used to accomplish this implements the Adjuinct Method. This is a direct method of inverting matrices. The indirect methods tried, such as elimination, proved unsatisfactory because of a large build-up of round-off error in the algorithm. The Adjuinct algorithm bounds the limit of round-off error. For further details see description of Subroutine INVERT.

(5) The Subroutines RANGE, THET, and PHI transform data represented in a X-Y-Z coordinate set to a R- θ - ϕ set. This is similar to the coordinate set described in Section 3, except that θ is replaced by $90^\circ - \phi$ and ϕ by $90^\circ + \phi$. It is possible for the user to assign his own coordinate system by replacing the three subroutines mentioned and the specified transformations of sensed position data in subroutine SENSOR.

(6) The plotting routines used to generate the graphs in the report are not listed. These routines were a package of special purpose routines available on Picatinny Arsenal's computer system. The user may incorporate his own routines in Entry COMP. This subroutine stores most of the important variables for the complete target aircraft flight path.

NARRATIVE DESCRIPTION

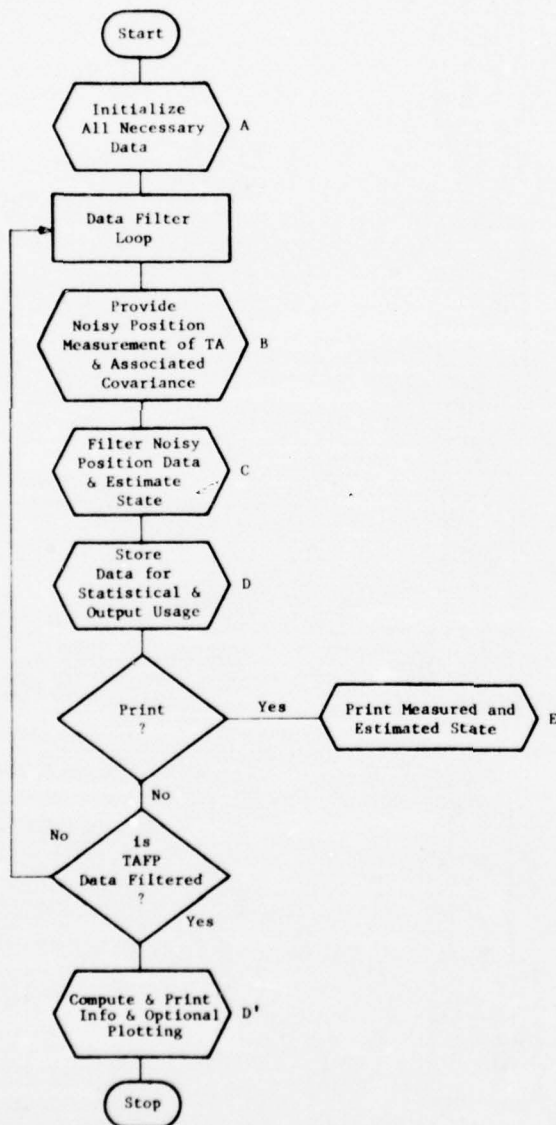
In order to begin the filtering process, the parameters of the filter must be specified (Subroutine INITIAL). It is also necessary to initialize the state variables as well as the corresponding error covariance matrix (Subroutine START). The simulated sensor (Subroutine SENSOR) begins operation with the "true" target aircraft state (Subroutine ACTUAL or Subroutine OWN) and adds white or correlated noise (Subroutine CORLATE and Subroutine RAND) to the state variables corresponding to position or position and velocity. The "true" target aircraft state is generated in an X-Y-Z coordinate set (Entry DATA or ENTRY COMP) and transformed to a R- θ - ϕ representation (Subroutines RANGE, THET, and PHI). Noise is added to the appropriate state variables in the R- θ - ϕ set, then transformed back to the X-Y-Z set. This output of the simulated sensor is an input to the Kalman filter, which provides an estimate of the present state of the target (Subroutines FILTER, MPRD, MTRA, and INVERT). The differences between the true and estimated state as well as the true and sensed state are computed (Subroutine COLLECT). This information is periodically printed (Subroutine OUTPUT) and constantly stored in memory. The mean square statistic is computed and printed based on the differences calculated along the entire target aircraft flight path (Subroutine STAT1).



PROGRAM FLOW AND ANALYSIS

All variables appearing in this section are listed within the subroutine in which they are defined. The variable list included in this section contains all program variables except those used for temporary storage.

Main Program XYZ



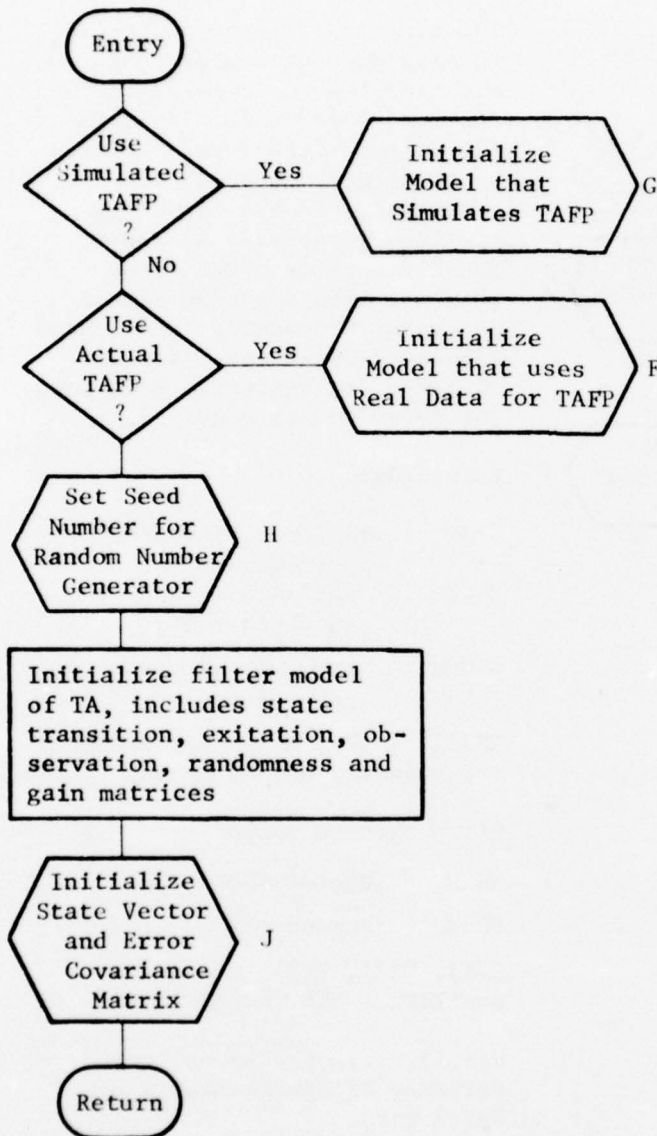
The function of the main program is to call the subroutines necessary to obtain the filter solution. The task and organization this routine is illustrated by the accompanying flow diagram.

Variables:

IPRINT = Specifies the frequency with which the filtered and "true" state are printed.

II = Do loop parameter for filtering

Subroutine INITIAL - A



This routine initializes all appropriate data and assigns values to user specified parameters. The model to generate the TAFP is chosen from two alternatives G&F on diagram (left). The matrices that appear in the Kalman filter equations are also initialized.

Variables:

L = number of target state variables

FLT = Determines which model is used to generate the TAFP (see comment cards in computer listings of Subroutine INITIAL).

SIG = estimated maximum acceleration squared of TA along either X, Y, or Z coordinate axis.

AHX, AHY, AHZ = a1, a2, a3

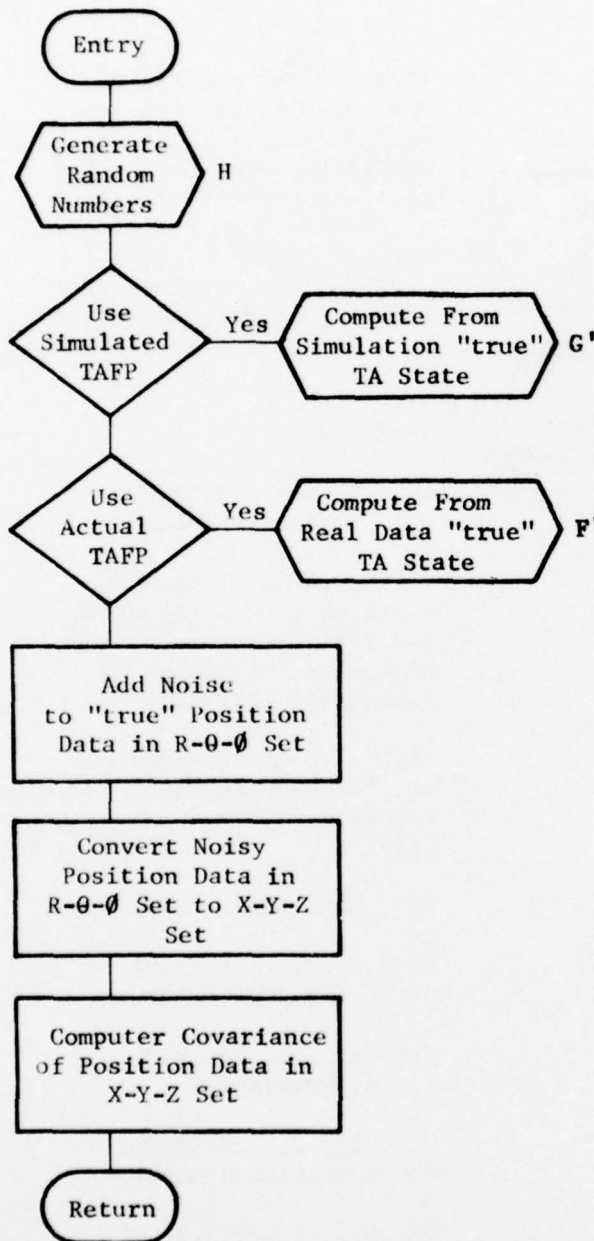
FH(1,1), ..., FH(9,9) = state transition matrix

H(1,1), ..., H(3,9) = state observation

GH(1,1), ..., GH(9,3) = state exitation matrix

Q(1,1), ..., Q(3,3) = state randomness covariance matrix

Subroutine SENSOR - B



The purpose of this routine is to simulate the sensor. An input necessary to the simulation is the "true" TA state. This is either generated from simulated or actual data as previously described. To the "true" TA position, expressed as $R-\theta-\phi$, the appropriate noise terms obtained from a random number generator are added. The sensed values of $R-\theta-\phi$ and their covariance are converted back into the X-Y-Z set as required.

Variables:

SIGR = variance of sensor
in range - R

SIGTH = variance of sensor
in theta - θ

SIGPH = variance of sensor in
phi - ϕ

XH(1), XH(2), XH(3) = independent
standard normal random variables

RS = sensed range

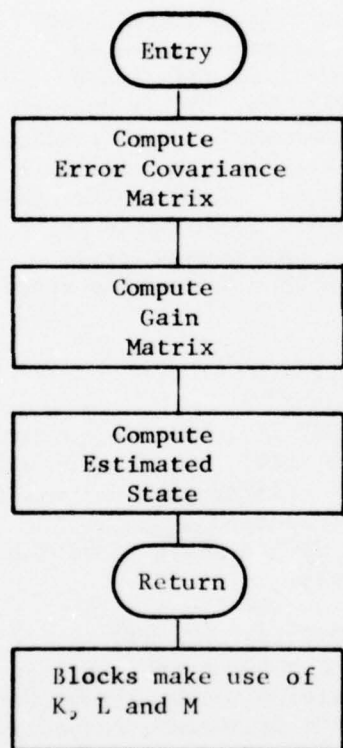
THSN = sensed theta

PHSN = sensed phi

Z(1), Z(2), Z(3) = sensed
position TA in X-Y-Z set

R(1,1), ..., R(3,3) = co-
variance of sensor matrix in
X-Y-Z set.

Subroutine FILTER - C



This routine implements the Kalman equations as described in Section 3. An input to this routine is noisy position measurements. The output is the estimate of TA state. In the filtering process, the P and K matrices are updated.

Execution of this routine requires call statements to MPRD and MTRA. It should be noted that these routines do not appear in the accompanying diagram.

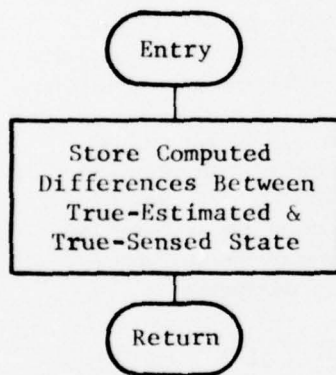
Variables:

$P(1,1), \dots, P(9,9)$ =
state error covariance matrix

$K(1,1), \dots, K(9,3)$ =
state gain matrix

$XI(1), \dots, XI(9)$ =
estimated state vector

Subroutine COLLECT - D



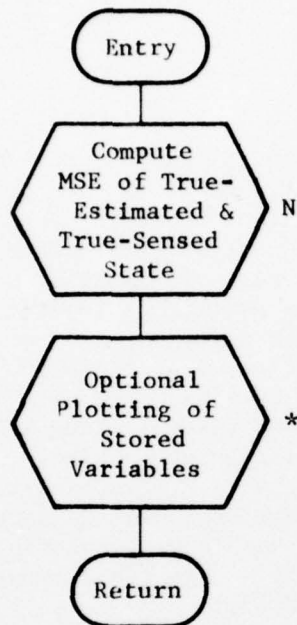
This routine computes and stores the difference between true and estimated state as well as the true and sensed state variables. As an option, the user may utilize the routine for statistical or special output purposes. Additional program variables may be accessed by expanding the argument lists of the call and Subroutine statements.

Variables:

DXP(II), DXV(II), DXA(II), DYP(II), DYV(II), DYA(II), DZP(II), DZV(II), DZA(II) = differences between the true and sensed position in the X-Y-Z set for the IIth iteration, respectively.

DXS(II), DYS(II), DZS(II) = differences between the true and sensed position in the X-Y-Z set for the IIth iteration, respectively.

Entry PLOTTING - D'



The second segment computes the MS statistic for the state estimation error and the sensor error. If optional storage of additional variables was utilized in the first segment, then any appropriate statistic could be generated.

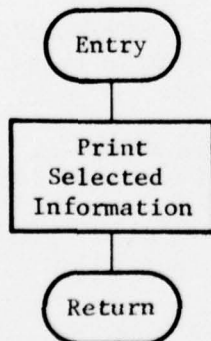
* At this point a plotting routine could be implemented by the user to plot selectively stored variables.

Variables:

SIG(1), ..., SIG(9) = the MSE for true - estimated state in the X, Y, Z direction for the entire TAFP, respectively.

SIG(10), SIG(11), SIG(12) = the MSE for the true - sensed state in the X, Y, Z direction for the entire TAFP, respectively.

Subroutine OUTPUT - E



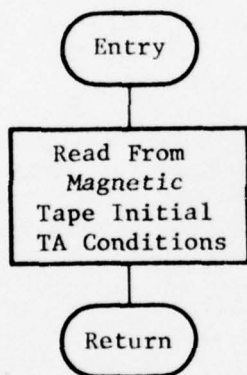
This routine provides an output for the printer. This includes printing the true state variables, estimated state variables, and errors in estimated state variables. The frequency of printing is controlled by the main program variable IPRINT.

Variables:

XXH(1), ..., XXH(9) = difference between true and estimated state

LABS, LABT, LABE, LABM = labels for output headings

Subroutine ACTUAL - F



This is a special purpose routine that generates the TAFP from real data. The first segment inputs the dummy header and the number of data records. The initial state of the TA is then inputted. The technique implemented in this routine is applicable only if the user has TAFP data in the form of initial state and a complete acceleration time history of the TA. If real data is available to the user in some other form, then he may construct his own model within the framework of this routine.

Variables:

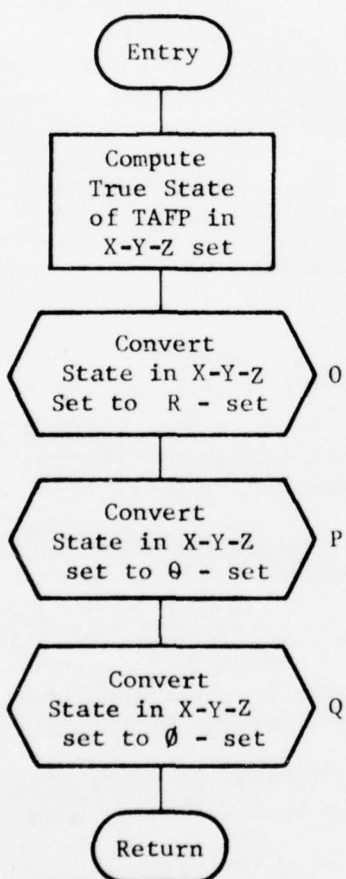
COM = dummy variable used to advance unnecessary records on magnetic tape.

NUMBER = total number of data records available for TAFP

NEND = number of data records containing states of TA that are to be filtered.

DT = actual time increment between records containing data

Entry DATA - F'



This second segment generates the TAFP given the initial state and the acceleration time history. This is accomplished as referenced in Section 2. The I th state is generated in an $R-\theta-\emptyset$ representation. The user may substitute his own coordinate set by replacing Subroutines RANGE, THET and PHI with his own routines.

Variables:

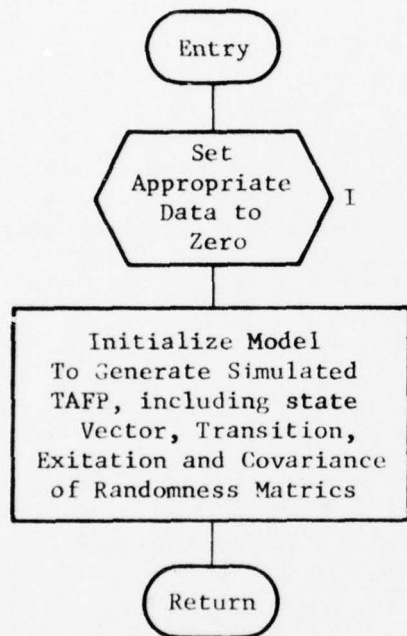
$X(1), \dots, X(9)$ = true state of TA in an X-Y-Z set

$RG(1), RG(2), RG(3)$ = TA state represented in R - set.

$TH(1), TH(2), TH(3)$ = TA state represented in θ - set.

$PHI(1), PHI(2), PHI(3)$ = TA state represented in \emptyset - set

$D(1), \dots, D(9)$ = true state vector of TA represented in an $R-\theta-\emptyset$ set



In this routine the TAFP is generated by implementing a model similar to the plant model used in constructing the Kalman filter equations. The difference between the two models arises from the choice of the parameters a_1 , a_2 , and a_3 . The first segment of the routine initializes all appropriate data and assigns values to user specified parameters.

Models other than those appearing in the body of the report may also be incorporated into this routine.

Variables:

$AX, AY, AZ = a_1, a_2, a_3$

$SIG =$ maximum acceleration squared of TA along either X, Y, or Z axis, respectively.

$DT =$ time difference between states generated by model for TAFP

$NEND =$ number of data records which will be generated, containing simulated state.

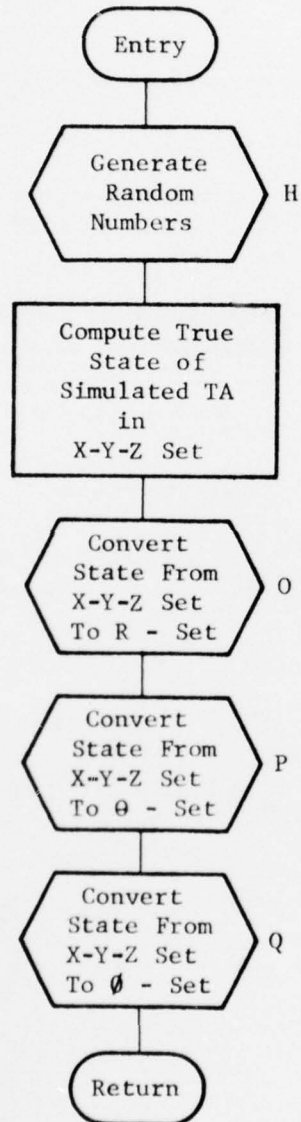
$X(1), \dots, X(9) =$ true state of TA in an X-Y-Z set.

$F(1,1), \dots, F(9,9) =$ state transition matrix.

$G(1,1), \dots, G(9,3) =$ state exitation matrix

$Q(1,1), \dots, Q(3,3) =$ state randomness covariance matrix

Entry COMP - G'



The second segment of this routine generates the Iith simulated TA state. This state is computed in an X-Y-Z set and then transformed to an R- Θ - \emptyset set.

Variables:

W(1), W(2), W(3) = U1, U2, U3.

D, TH, RG, and PHI are the same variables as described in F.

Subroutine RAND - H

This routine generates a sample of random variables from either a population which is normally distributed with mean zero and variance one or from a population uniformly distributed on the closed interval zero to one. It was taken from the LOFAADS model developed by Litton Industries for the Frankford Arsenal.

Initialization:

N = 0 (Integer)

X(1) = Five digit positive whole number to be used as a seed number (Real).

Actual Usage:

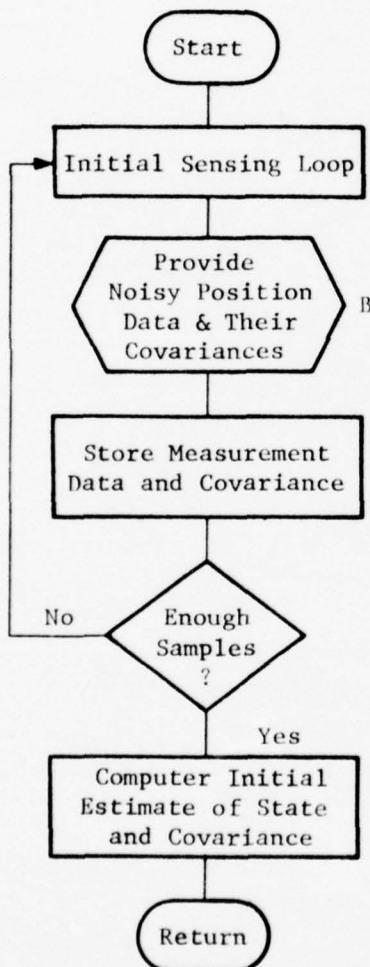
N = integer number ($\neq 0$) which determines the number of identically independent random variables that are generated. $N < 0$ provides uniform random variables, $N > 0$ provides normal random variables.

X(1), ..., X(|N|) = an array of length "N" containing the specified type of random variables ("N" is restricted only by the Dimension of X).

Subroutine ZERO - I

This routine sets to zero a specified matrix or a specified vector. The computer listings at the end of this Appendix for this routine are self-explanatory.

Subroutine START - J



This routine initializes the target state vector and the corresponding error covariance matrix. The initialization scheme is described in Section 4.

Variables:

$I1 = 1$

$I2 =$ number of initial measurements to be made in order to initialize the estimated state and its covariance.

$I =$ Do loop parameter of the initialization loop.

$ZV(1,1), \dots, ZV(3,I2) =$ stores the measurements of the sensor for the I th iteration.

$V(1, 1, 1), \dots, V(3, 3, I2) =$ stores the covariance of the sensor for the I th iteration.

$XH(1), \dots, XH(9) =$ estimated state vector of TA.

$P(1,1), \dots, P(9,9) =$ state error covariance matrix.

Subroutine MPRD - K

This routine multiplies two properly defined matrices. The computer listing for this routine at the end of this appendix is self-explanatory.

Subroutine MTRA - L

This routine transposes a matrix. The computer listing for this routine at the end of this appendix is self-explanatory.

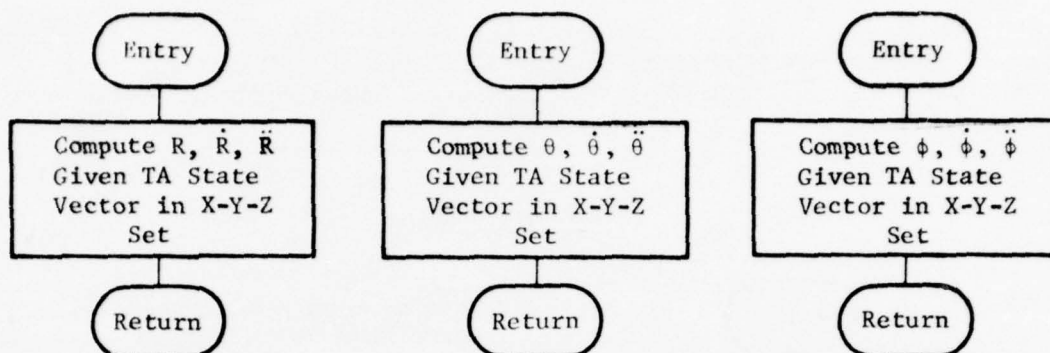
Subroutine INVERT - M

This is a special purpose routine used to invert a 3×3 matrix in the most efficient manor. It employs a direct method known as the Adjoint algorithm. The inverse obtained is for all practical purposes exact, because the algorithm employs a test on the determinant of the matrix. This test compares the value of the determinant with a specified value (in this case .01). If the determinant is less than this value the complete program stops. Thus, the possibility of round-off error due to "near singular" matrices is eliminated.

Subroutine STAT1 - N

This routine computes the mean square statistic of a given array.

Subroutines RANGE - θ , THET - P, PHI - Q



These three routines together convert the TA state vector from an X-Y-Z coordinate set to an R- θ - ϕ coordinate set.

Variables:

D(1), ..., D(9) = TA state vector represented in an X-Y-Z set.

$$O \left\{ \begin{array}{l} X(1), X(2), X(3) = R, \dot{R}, \ddot{R} \\ R = \text{range} \\ RDOT = \text{range rate} \\ RDDOT = \text{range acceleration} \end{array} \right.$$

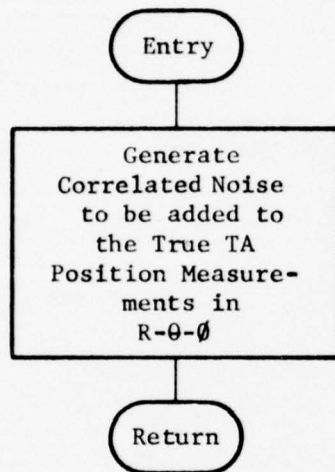
$$P \left\{ \begin{array}{l} N = \text{this integer counts the number of times the angle } \theta \text{ changes sign. The desired representation for } \theta \text{ is a positive angle, but the ATAN2 Fortran function is defined for } -\pi < \theta < \pi. \text{ Thus, } \theta \text{ is transformed to } 0 < \theta < 2 * N * \pi. \\ X(1), X(2), X(3) = \theta, \dot{\theta}, \ddot{\theta} \\ THETA = \text{theta} \\ TDOT = \text{angular rate of theta} \\ TDDOT = \text{angular acceleration of theta} \end{array} \right.$$

$$Q \left\{ \begin{array}{l} X(1), X(2), X(3) = \phi, \dot{\phi}, \ddot{\phi} \\ PHII = \text{phi} \\ PDOT = \text{angular rate of phi} \\ PDDOT = \text{angular acceleration of phi} \end{array} \right.$$

OPTIONAL FILTERING AND SENSING ALGORITHMS

The modifications of, and additions to the complete computer program are presented. This permits the user to simulate a variety of filtering and sensing algorithms utilizing the basic form of the previously described computer program.

Subroutine CORR



This routine generates correlated noise that is added to the true TA position variables to provide the simulated sensor measurements. The input to this routine is a vector of size 3 that is composed of independent normally distributed random variables with zero mean and variance one. The output is a vector of time correlated noise.

Since this is an optional routine, it must be inserted into the complete computer program. There are some modifications to Subroutine SENSOR which are required. The statements in SENSOR corresponding to noise terms being added to the true position are to be replaced with

```
RS = X(1) + COR (1)
THS = X(2) + COR (2)
PHS = X(3) + COR (3)
```

and an additional statement just before this substitution is to be

```
CALL CORR (COR, XN).
```

As seen above COR is a three element array. It must be dimensioned in storage by

```
COMMON/CORRL/COR(3)
```

corresponding to an identical statement in Subroutine CORR.

Variables:

CORTM = correlation time (assumed the same for R, θ , ϕ measurements).

DT = time increment between measurements (same as DT in other routines).

ALP = reciprocal of correlation time.

SIGR = variance of range measurements.

SIGTH = variance of theta measurements.

SIGPH = variance of phi measurements.

COR(1), COR(2), COR(3) = correlated noise in R, θ , ϕ set.

Rate Measurements

The addition of rate measurements \dot{R} , $\dot{\theta}$, and $\dot{\phi}$ to a position only sensor can be easily accomplished. The necessary modifications involve augmenting selected matrices and restructuring certain subroutines. The changes are the following:

Matrices:

- 1) H - observation matrix
- 2) R - covariance of sensor matrix
- 3) Z - sensor matrix
- 4) K - gain matrix

Wherever in the complete listings Z, H, R and K appear dimensioned in storage, they must be changed to Z(6), H(6,9), R(6,6) and K(9,6).

Subroutines:

INITIAL

The call to Subroutine ZERO that contains K and H must be changed to the proper dimension.

The new H matrix is defined by $H(i, i) = 1, i = 1, \dots, 6$ and $H(i, j) = 0, i \neq j$.

The Call to Subroutine START as well as Subroutine START may be eliminated since the object of START is to obtain an estimate of velocity and its covariance. The initial estimate of TA state is composed of the sensor measurements. The elements of the initial state estimation covariance matrix consists of the variance of the sensor measurement expressed in the X-Y-Z coordinate set. The initial estimate of acceleration remains zero with error constrained by the physical limitation of TA which is the Fortran variable SIG.

The K matrix is augmented so that $K(4, 4) = K(5, 5) = K(6, 6) = .5$ which ignores cross terms. This is a good approximation that is quickly corrected in succeeding iterations of the filter equations.

FILTER

In addition to the modification in the dimension of the previously stated variables, the variables HT(9,6), KT(6,9), PHT(9,6), HPHT(6,6), RP(6,6) and RHI(6,6) are to be dimensioned in storage as shown here. Also, within executable statements where these variables appear, the numerical three's are to be replaced by numerical sixes.

There is a Call to Subroutine INVERT. The routine which appears in the computer listings is a special routine that only inverts a 3 x 3 matrix. The user must provide his own routine to invert a 6 x 6 matrix.

SENSOR

In addition to the modification in the dimension of the previously stated variables, the variable XN(6) is dimensioned in storage as shown here. The rate sensor covariance terms used in the report are as follows:

```
SIGRD  = 25.
SIGTHD = SIGPHD = .25 E-06
SPHPTH = SIGTH + SIGPH
SDPHTH = SIGTHD + SIGPHD
```

The new sensed \dot{R} , $\dot{\theta}$, $\dot{\phi}$ are defined by:

```
RDS = D(2) + XN(4) * SQRT(SIGRD)
THDS = D(5) + XN(5) * SQRT(SIGTHD)
PHDS = D(8) + XN(6) * SQRT(SIGPHD)
```

The additional sensor measurements in an X-Y-Z coordinate set are:

```
Z(4) = -RDS * COS (PHS) * SIN (THS) + RS * SIN (THS) * SIN (PHS)
* PHDS
Z(5) = RDS * COS (THS) * COS (PHS) - RS * SIN (PHS) * COS (THS)
* PHDS - RS * COS (PHS) * SIN (THS) * THDS
Z(6) = RDS * SIN (PHS) + RS * COS (PHS) * PHDS
```

The R matrix now becomes a 6 x 6 matrix with the following additional terms:

```
RDS2 = RDS * RDS
THDS2 = THDS * THDS
PHDS2 = PHDS * PHDS
R(4,4) = (RDS2 + SIGRD) * (SIGTH * TC2 * PC2 + SIGPH * PS2 * TS2 +
SIGTH * SIGPH * PS2 * T(2) + (SIGTH * PS2 * TC2 + SIGPH * PC2 * TS2 + SIGTH
* SIGPH * PC2 * TS2) + RS2 * SIGPHD * PS2 * TS2 + RS2 * (THDS2 + SIGTH) *
(SIGTH * PC2 * TS2 + SIGPH * PS2 * TC2 + SIGTH * SIGPH * PS2 * TS2) +
RS2 * SIGTHD2 * PC2 * TC2 + 2. * RS * RDS * (THDS * SIGTH * PC2 * TS2
- PHDS * SIGPH * PHC * PHS * THC + SIGPH * THDS * PS2 * THS) * PHC

R(5,5) = (RDS2 + SIGRD) * (SIGTH * PC2 * TS2 + SIGPH * PS2 * TC2
SIGTH * SIGPH * PS2 * TS2) + SIGRD * PC2 * TC2 + RS2 * (PHDS2 + SIGPHD)
* (SIGTH * PS2 * TS2 + SIGPH * TC2 * PC2 + SIGTH * SIGPH * PC2 * TS2) +
RS2 * SIGPHD * PS2 * TC2 + RS2 * (THDS2 + SIGTHD) * (SIGPH * PC2 * TC2 +
SIGPH * PS2 * TS2 + SIGTH * SIGPH * PS2 * TS2) + RS2 * SIGTHD * PC2 *
TS2 - 2. * RS * RSD * (THDS * SIGTH * PC2 * THC + PHDS * SIGTH * PHC *
PHS * THS - SIGPH * THDS * PS2 * THC) * THS
```

$$R(6,6) = RS2 * PC2 + RS2 * (PHDS2 + SIGPH) * SIGPH * PS2 + SGR * PHDS2 * PC2 + (RDS2 + SIGRD) * SIGPH * PC2 + SIGRD * PS2 - 2. * RS * RDS * PHDS * SIGPH * PHC * PHS$$

$$R(1,4) = RS * RDS * (SIGTH * PC2 * TC2 + SIGPH * PS2 * TS2 + SIGTH * SIGPH * PS2 * TC2) - PHDS * SIGTH * RS2 * PHC * PHS * TC2 + THDS * SIGPH * RS2 * PS2 * THC * THS$$

$$R(2,5) = RS * RDS * (SIGTH * PC2 * TS2 + SIGPH * TS2 * PC2 + SIGTH * SIGPH * PS2 * TS2) + PHDS * SIGTH * RS2 * PHC * PHS * TS2 + THDS * SIGPH * RS2 * PS2 * THS * THC$$

$$R(3,6) = RS * RDS * PC2$$

The elements $R(4,1)$, $R(5,2)$, $R(6,3)$ are equivalent to $R(1,4)$, $R(2,5)$, $R(3,6)$ since R is a symmetric matrix. The $R(1,5)$, $R(1,6)$, $R(2,4)$, $R(2,6)$, $R(3,4)$ and $R(3,5)$ elements are not given, but may be easily derived by the user. Elements not described are set to zero.

COLLECT

In addition to the modification in the dimension of the previously stated variables, the new variables $DXSV(300)$, $DYSV(300)$, $DZSV(300)$ must be inserted. Also, $SIG(15)$ is to be dimensioned in storage as shown here.

The differences between the true velocity and the sensor measurement of velocity is computed as:

$$\begin{aligned} DXSV(II) &= X(4) - Z(4) \\ DYSV(II) &= X(5) - Z(5) \\ DZSV(II) &= X(6) - Z(6) \end{aligned}$$

Entry PLOTTING

The mean squared velocity error is computed and printed by the following:

```
CALL STAT1 (DXSV, N, SIG(13))
CALL STAT1 (DYSV, N, SIG(14))
CALL STAT1 (DZSV, N, SIG(15))
PRINT 3000, (SIG(I), I = 10, 15)
```

Uncoupled Filter

The plant model for the TA utilized by the Kalman filter is constructed with state variables represented in an X-Y-Z coordinate set. The triplet of state variables (position, velocity and acceleration) for each coordinate axis is assumed to be independent of triples with respect to other axis. However, the X-Y-Z coordinate set is not compatible with the R- θ - ϕ set of the sensor measurements which are assumed to be independent of each other. In order to make the sensor data compatible with the filter equations, the independent R- θ - ϕ measurements are transformed to dependent X-Y-Z measurements. The dependence is reflected in the non-zero off-diagonal elements of the R matrix. The uncoupled three state filters can now be obtained from the nine state filter by setting the off diagonal elements of the R matrix equal to zero.

Similar problems are inherent in the rate aided sensor, except that there are three additional rate measurements. Again, the sensor variables are transformed into the X-Y-Z set. By setting the off diagonal elements in the resulting R matrix equal to zero, the user obtains the decoupled three state filters.

Steady State Filter

Having uncoupled the nine state filter by setting the off diagonal elements in the R matrix to zero, the user must now replace the remaining diagonal elements with constant values. This eliminates the dependence of the filter equations on the target state. These elements in the R matrix are assigned the following values.

Position only sensor model:

$$R(i, i) = \text{SIGR} \quad i = 1, 2, 3$$

Rate aided sensor model:

$$R(i, i) = \text{SIGRD} \quad i = 4, 5, 6$$

To obtain steady state results the entire computer program XYZ must be processed twice; the first time using constant values in the R matrix (as stated) that generate steady state gain values (K matrix). The second time, using these steady state gains to compute the estimated state along the entire TAFP. This substitution considerably reduces total computer time and costs. This is accomplished because the entire filtering routine is replaced by the following statements in the main program.

```
CALL MPRD (K, H, KH, L, 3, L)
CALL MPRD (FH, XH, FHX, L, L, 1)
DO 100 I = 1, L
DO 100 J = 1, L
IF(I.EQ.J) GO TO 95
KH(I,J) = -KH(I,J)
```

```

      GO TO 100
95  CONTINUE
      CALL MPRD (K, Z, KZ, L, 3, 1)
      CALL MPRD (KH, FHX, KHFX, L, L, 1)
      DO 110 I = 1, L
110  XH(I) = KHFX(I) + KZ(I)

```

These statements are positioned in the main program in place of Call FILTER. The variable designation and dimension have to be modified to insure a legal substitution in the main program.

```

      IMPLICIT REAL (K)
      DIMENSION KH (9,9), FHX(9), KZ(3), KHFX(9)
      L = 9

```

The steady state rate aided filter is similar to the one necessary to obtain the position only steady state filter. The procedure described above carries over to this case except that all numerical threes appearing in the proceeding program statements are to be replaced with numerical sixes.

Combined Algorithms

All possible combinations of filter and sensor algorithms have not been presented in this report. For example, the user might wish to observe the effect of correlated noise in a steady state rate filter. This may be accomplished with minimum alteration to the major components of the program because of its modular design. However, the user must be very careful when implementing changes, especially in the alteration of the dimension or complete removal of program variables.

INDEX TO ROUTINES

This index provides the following information for the main program and the subroutines where applicable.

- 1) Alphabetic indicators that appear in subroutine flow charts
- 2) Name of routine
- 3) Page number in this appendix where routine description appears
- 4) Computer memory storage requirement represented in decimal words
- 5) Argument list - calling and returning
 - a. specific list - these routines were written for the listed variables
 - b. general list - these routines were written for any suitable variables

Letter	Name of Routine	Page	Memory	Calling Variables	Returning Variables
--	XYZ	111	3476	--	--
A	INITIAL	112	157	X, XH, FH, GH, P, K, Q, H, NEND, OTIME	Same except DT = DTIME
B	SENSOR	113	28	Z, X, R, D	Same
C	FILTER	114	911	Z, R, Q, ZH, GH, H, K, FH, P	Same
D	COLLECT	115	7261	X, XH, Z, II, DTIME, N	Same
D'	PLOTTING	116	--	Same as Above	--
E	OUTPUT	117	105	X, XH, P, K, TIME, II	Same
F	ACTUAL	118	55	Same as Above	--
G	OWN	120	241	X, D, DT, NEND	Same
G'	COMP	121	--	Same as Above	--
H	RAND*	122		Length of Array, Array	N, X
I	ZERO	123	7	Matrix, No. of Rows, No. of Col.	A, I, J
J	START	123	91	Z, X, XH, P, R, DT	Same except DT = DTIME
K	MPRD	124		M1**, M2, M1 * M2, No. of rows in 1 & 2, No. of Cols in 1 and rows in 2, No. of Cols in 1 & 2	A, B, R, N, M, L
L	MTRA	124		M1, transpose of M1, No. of rows in 1, No. of Cols in 1	A, B, R, N, M, L
M	INVERT	124	9	R, RI	A, AI

<u>Letter</u>	<u>Name of Routine</u>	<u>Page</u>	<u>Memory</u>	<u>Calling Variables</u>	<u>Returning Variables</u>
N	STAT1	124	8	Array, Length of Array	D, N, SIG
O	RANGE	125	11	X, RG	D, X
P	THET	125	18	X, TH	D, X
Q	PHI	125	16	X, PH	D, X

* Call made after initialization

** M = Matrix

' = Entry into subroutine

COMPUTER PROGRAM LISTING

```

000000000000000000000000
PROGRAM XYZ(INPUT,OUTPUT,TAPES)
REAL *
COMMON /ZLIM/ I
DIMENSION Z(3),X(9),XH(9),FH(9,9),GH(9,3),P(9,9),K(9,3),R(3,3)
1,H(3,9),Q(3,3),D(9)
C INITIALIZE PRINT FREQUENCY
DATA IPRINT/50/
CALL INITIAL(X,XH,FH,GH,P,K,Q,H,NEND,DTIME)
C START DATA FILTERING LOOP
DO 10 II=1,NEND
TIME=FLOAT(II)*DTIME
LMOD=II/IPRINT
LMOD=LMOD*IPRINT+1
CALL SENSOR(Z,X,R,D)
CALL FILTER(Z,P,Q,XH,GH,H,K,FH,P)
CALL COLLECT(X,XH,Z,II,DTIME,NEND)
IF(LMOD.EQ.II)
1CALL OUTPT(X,XH,P,K,TIME,II)
10 CONTINUE
CALL PLOTTING(X,XH,Z,II,DTIME,NEND)
STOP
END
SUBROUTINE INITIAL(X,XH,FH,GH,P,K,Q,H,NEND,DT)
REAL *
COMMON /ZLIM/ I
COMMON /ZMOD/ I
DIMENSION Z(3),X(9),XH(9),FH(9,9),GH(9,3),P(9,9),K(9,3),R(3,3)
DIMENSION D(9),ALPHA(1),H(3,9),Q(3,3)
DATA AHX,AHY,AHZ/30.,1/
C SPECIFY NUMBER OF STATE S TO BE ESTIMATED L=
L=4
C SPECIFY FLT=THOWN FOR COMPUTER SIMULATED FLIGHT DATA DATA
C (SEE SUBROUTINE OWN)
C SPECIFY FLT=14 FOR ACTUAL FLIGHT DATA DATA (SEE SUBROUTINE ACTUAL)
FLT=THOWN
FLT=THOWN
SIG=1000.
PRINT 1000
1000 FORMAT(1H)
IF(FLT.NE.THOWN) CALL ACTUAL(X,D,DT,NEND)
IF(FLT.EQ.THOWN) CALL OWN(X,D,DT,NEND)
RX=2.*AHX*SIG
RY=2.*AHY*SIG
RZ=2.*AHZ*SIG
C INITIALIZE RANDOM NUMBER GENERATOR
CALL RAND(0,37*12.)
C SET ZERO S
CALL ZERO(K,1,3)
CALL ZERO(Q,3,3)
CALL ZERO(GH,1,3)
CALL ZERO(P,1,1)
CALL ZERO(FH,1,1)
CALL ZERO(H,3,1)
C KALMAN FILTERS --PLANT MODEL
C INITIALIZE TRANSITION MATRIX--FH
TEMPX=EXP(-AHX*DT)
TEMPY=EXP(-AHY*DT)
TEMPZ=EXP(-AHZ*DT)
TEMAX=(1.-TEMPX)/AHX
TEMAY=(1.-TEMPY)/AHY
TEMAZ=(1.-TEMPZ)/AHZ
FH(1,1)=1.
FH(2,2)=1.
FH(3,3)=1.
FH(4,4)=1.

```

```

      FH(6,6)=1.
      FH(7,7)=TFMAX
      FH(1,4)=DT
      FH(1,7)=(DT-TFMAX)/AHX
      FH(4,7)=TFMAX
      FH(5,5)=1.
      FH(8,8)=TFMDY
      FH(2,5)=DT
      FH(2,8)=(DT-TFMAX)/AHY
      FH(5,8)=TFMAX
      FH(9,9)=TFMPZ
      FH(3,6)=DT
      FH(3,9)=(DT-TFMAX)/AHZ
      FH(6,9)=TFMAX
C      INITIALIZE OBSERVATION MATRIX-- H
      H(1,1)=1.
      H(2,2)=1.
      H(3,3)=1.
C      INITIALIZE EXCITATION MATRIX-- GH
      GH(1,1)=(DT*DT/2.-DT/AHX*TFMAX/AHX)/AHX
      GH(2,2)=(DT*DT/2.-DT/AHY*TFMAX/AHY)/AHY
      GH(3,3)=(DT*DT/2.-DT/AHZ*TFMAX/AHZ)/AHZ
      GH(4,1)=(DT-TFMAX)/AHX
      GH(5,2)=(DT-TFMAX)/AHY
      GH(6,3)=(DT-TFMAX)/AHZ
      GH(7,1)=TFMAX
      GH(8,2)=TFMAX
      GH(9,3)=TFMAX
C      INITIALIZE STATE RANDOMNESS COVARIANCE MATRIX --Q
      Q(1,1)=RX/DT
      Q(2,2)=RY/DT
      Q(3,3)=RZ/DT
C      INITIALIZE STATE VECTOR-- XH AND ERROR COVARIANCE MATRIX-- P
      CALL START(Z,X,XH,P,R,DT)
      P(7,7)=SIG
      P(8,8)=SIG
      P(9,9)=SIG
C      INITIALIZE GAIN MATRIX-- K
      K(1,1)=.5
      K(2,2)=.5
      K(3,3)=.5
200 FORMAT(9(1X,F12.6,1X))
100 FORMAT(1(1X,3(1X,F12.6,1X),/))
      PRINT 500
500 FORMAT(///.* *)
      PRINT 400,AHX,AHY,AHZ,SIG
400 FORMAT(15X*THIS IS A 9 STATE KALMAN FILTERED SIMULATION -ALL VARIABLE
      S ARE IN M-K-S UNITS*,//,10X,*OMEGA X = *,F6.3,5X,*OMEGA Y = *,
      F6.3,5X,*OMEGA Z = *,F6.3,5X,*SIGMA = *,F6.2,/)
      PRINT 105
105 FORMAT(/,10X,*TRANSITION MATRIX-- FH      *)
      PRINT 200,FH
      PRINT 108
108 FORMAT(/,10X,*EXCITATION MATRIX --GH*)
      PRINT 200,GH
      PRINT 104
104 FORMAT(/,10X,*ESTIMATED STATE VECTOR--XH*)
      PRINT 100,XH
      PRINT 106
106 FORMAT(/,10X,*STATE ERROR COVARIANCE MATRIX-- P*)
      PRINT 200,P
      PRINT 103
103 FORMAT(/,10X,*GAIN MATRIX-- K*)
      PRINT 100,K
      RETURN
      END

```

```

SUBROUTINE START(Z,X,XH,P,P,DTIME)
DIMENSION Z(3),XH(9),P(9,9),R(3,3),V(3,3,4),ZV(3,4)
DIMENSION D(9),X(9)
DATA 11/1/.12/4/
PRINT 1000
1000 FORMAT(/,10X,'INITIAL SENSOR MEASUREMENTS',25X,'VARIANCES OF THESE
      1 MEASUREMENTS',/ )
      BEGIN SENSOR MEASUREMENT AND ERROR COVARIANCE STORAGE
      DO 10 I=1,12
      CALL SENSOR(Z,X,XH,P,D)
      V(1,1,1)=P(1,1)
      V(2,2,1)=P(2,2)
      V(3,3,1)=P(3,3)
      V(1,2,1)=P(1,2)
      V(1,3,1)=P(1,3)
      V(2,3,1)=P(2,3)
      V(2,1,1)=P(1,2)
      V(3,1,1)=P(1,3)
      V(3,2,1)=P(2,3)
      ZV(1,1)=Z(1)
      ZV(2,1)=Z(2)
      ZV(3,1)=Z(3)
10 PRINT 2000,(Z(I),I=1,3),V(1,1,1),V(2,2,1),V(3,3,1)
2000 FORMAT(3(2X,F14,6),8X,3(2X,F14,6))
      DIF=ELDAT(12-11)*DTIME
      DT2=DIF*DIF
      INITIALIZE ESTIMATED STATE VECTOR-- XH
      DO 30 I=1,3
      XH(I)=ZV(I,1)
      XH(I,3)=(ZV(I,12)-ZV(I,11))/DIF
      INITIALIZE ESTIMATED STATE ERROR COVARIANCE MATRIX-- P
      P(1,1)=V(1,1,1)
30 P(1,3,1)=(V(1,1,12)+V(1,1,11))/DT2
      P(1,2)=V(1,2,1)
      P(1,3)=V(1,3,1)
      P(2,3)=V(2,3,1)
      P(2,1)=P(1,2)
      P(3,1)=P(1,3)
      P(3,2)=P(2,3)
      DO 40 I=1,3
      DO 40 J=1,3
      IF (I.EQ.J) GO TO 45
      P(I,3,J,3)=(V(I,1,12)+V(I,1,11))/DT2
40 CONTINUE
      RETURN
      END
SUBROUTINE FILTER(Z,Q,Q,XH,GH,H,K,FK,P)
IMPLICIT REAL(K)
COMMON /LIM/ L
DIMENSION X(9),XH(9),H(3,9),HT(9,3),K(9,3),KT(3,9),Z(3)
1 .FH(9,9),FHT(9,9),P(9,9),GH(9,3),GT(3,9),GG(9,9),R(3,3),Q(3,3)
2 .KHP(9,9),FKHP(9,9),KHT(9,9),GHQ(9,3),KR(9,3),FHR(9,3)
3 .FKRKT(9,9),KKT(9,9),PHT(9,3),HPHT(3,3),RP(3,3),RPT(3,3)
4 .FHX(9),KZ(9),FHFHX(9),KH(9,9)
      REGINS COMPUTATION OF ESTIMATED STATE ERROR COVARIANCE MATRIX-- P
      CALL MPRD(K,H,KH,L,3,L)
      DO 50 I=1,L
      DO 50 J=1,L
      IF (I.EQ.J) GO TO 60
      KH(I,J)=-KH(I,J)
      GO TO 50
50 KH(I,J)=1.-KH(I,J)
60 CONTINUE
      CALL MTRA(KH,KHT,L,L)
      CALL MPRD(KH,P,KHP,L,L,L)
      CALL MPRD(FH,KHP,FKHP,L,L,L)

```

```

      CALL MTRA(FH,FHT,L,L)
      CALL MPRD(FKHP,KHT,KHP,L,L,L)
      CALL MPRD(KHP,FHT,KH,L,L,L)
      CALL MTRA(GH,GT,L,3)
      CALL MPRD(GH,G,GH,L,3,3)
      CALL MPRD(GH,GT,G,L,3,L)
      CALL MTRA(K,KT,L,3)
      CALL MPRD(K,Z,KR,L,3,3)
      CALL MPRD(FH,KR,FHR,L,L,3)
      CALL MPRD(FH,KR,FKR,L,L,3)
      CALL MPRD(FH,KR,FKR,L,L,3)
      CALL MPRD(FKHT,FHT,FKT,L,L,L)
      DO 90 I=1,L
      DO 90 J=1,L
      90 P(I,J)=KH(I,J)+GG(I,J)+KKT(I,J)
C      ENDS COMPUTATION OF P-MATRIX
C      BEGINS COMPUTATION OF GAIN MATRIX-- K
      CALL MTRA(H,HT,L,3)
      CALL MPRD(P,HT,PHT,L,L,3)
      CALL MPRD(H,PHT,HPHT,L,3,3)
      DO 90 I=1,3
      DO 90 J=1,3
      90 RP(I,J)=HPHT(I,J)+P(I,J)
      CALL INVERT(RP,RP)
      CALL MPRD(PHT,RP,K,L,3,3)
C      ENDS COMPUTATION OF K-MATRIX
C      BEGINS COMPUTATION OF ESTIMATED STATE VECTOR-- XH
      CALL MPRD(K,H,KH,L,3,L)
      CALL MPRD(FH,XH,FHX,L,L,3)
      DO 100 I=1,L
      DO 100 J=1,L
      IF(I.EQ.J) GO TO 95
      KH(I,J)=-KH(I,J)
      GO TO 100
      95 KH(I,J)=1.-KH(I,J)
      100 CONTINUE
      CALL MPRD(K,Z,KZ,L,3,1)
      CALL MPRD(KH,FHX,KHFHX,L,L,3)
      DO 110 I=1,L
      110 XH(I)=KHFHX(I)+KZ(I)
C      ENDS COMPUTATION OF XH-VECTOR
      RETURN
      END
      SUBROUTINE SENSOR(Z,X,P,D)
      COMMON /WHO/ FLT
      DIMENSION X(9),D(2),Z(3),XN(3),R(3,3)
C      SENSOR ACCURACIES (VARIANCES)
      DATA SIGR,SIGTH,SIGPH/100.,2*1E-06/
      CALL RAND(3,XN)
C      LOGICAL DECISION AS TO WHICH METHOD OF TARGET AIRCRAFT
C      FLIGHT PATH GENERATION IS TO BE IMPLEMENTED
C      SEE SUBROUTINE FOR DETAILS
      IF (FLT.EQ.3) THEN 1 CALL COMPT(X,D,DT,NEND)
      IF (FLT.NE.3) THEN 2 CALL DATA(X,D,DT,NEND)
C      GENERATION OF SENSED VALUES
      RS=D(1)+XN(1)*SQRT(SIGR)
      THSN=D(4)+XN(2)*SQRT(SIGTH)
      PHSN=D(7)+XN(3)*SQRT(SIGPH)
      THS=SIN(THSN)
      THC=COS(THSN)
      PHS=SIN(PHSN)
      PHC=COS(PHSN)
C      COMPUTE SENSED STATE VECTOR-- Z
C      NOTE-- IF USER PROVIDES HIS OWN COORDINATE SET
C      THEN THE FOLLOWING TRANSFORMATIONS MUST ALSO BE
C      MODIFIED
      Z(1)=-RS*PHC*THS

```



```

      Z(2)=RS*PHC*THC
      Z(3)=RS*PHS
C      COMPUTE COVARIANCE OF SENSED STATE MATRIX-- R
      R2=RS*RS
      TS2=THS*THS
      TC2=THC*THC
      PS2=PHS*PHS
      PC2=PHC*PHC
      R(1,1)=R2*(SIGPH*PS2*TS2+SIGTH*SIGPH*PS2*TC2+SIGTH*PC2*TC2)
1      +SIGR*PC2*TS2
      R(2,2)=R2*(SIGPH*PS2*TC2+SIGTH*SIGPH*PS2*TS2+SIGTH*PC2*TS2)
1      +SIGR*PC2*TC2
      R(3,3)=R2*SIGPH*PC2+SIGR*PS2
      R(1,2)=R2*TS2*TC2*(SIGPH*PS2-SIGTH*PC2)-SIGR*PC2*THC*THS
      R(1,3)=- (SIGR-R2*SIGPH)*PHS*PHC*THS
      R(2,3)=(SIGR-R2*SIGPH)*PHS*PHC*THC
      R(2,1)=R(1,2)
      R(3,1)=R(1,3)
      R(3,2)=R(2,3)
      RETURN
      END
      SUBROUTINE ACTUAL(X,D,DT,NEND)
      DIMENSION X(9),D(9)
1      RG(3),TH(3),PH(3)
      REWIND 5
      READ(5,10) COM
10      FORMAT(A1)
      READ(5,20) NUMBER
20      FORMAT(I4)
C      SPECIFY LENGTH(NEND) AND TIME INCREMENT(DT) OF TARGET
C      FLIGHT PATH GENERATION
      NEND=NUMBER-5
      DT=.1
      DT2=DT*DT/2.
C      INPUT FROM TAPE INITIAL TRUE STATE CONDITIONS
      READ(5,30) (X(I),I=1,9)
      RETURN
      ENTRY DATA
C      TRUNCATION OF SERIES EXPANSION TO OBTAIN POSITION,
C      VELOCITY FROM ACCELERATION DATA
      X(1)=X(1)+X(4)*DT+X(7)*DT2
      X(2)=X(2)+X(5)*DT+X(8)*DT2
      X(3)=X(3)+X(6)*DT+X(9)*DT2
      X(4)=X(4)+X(7)*DT
      X(5)=X(5)+X(8)*DT
      X(6)=X(6)+X(9)*DT
C      INPUT FROM TAPE ACCELERATION DATA
      READ(5,40) X(7),X(8),X(9)
      CALL RANGE(X,RG)
      CALL THET(X,TH)
      CALL PHI(X,PH)
      D(1)=RG(1)
      D(2)=RG(2)
      D(3)=RG(3)
      D(4)=TH(1)
      D(5)=TH(2)
      D(6)=TH(3)
      D(7)=PH(1)
      D(8)=PH(2)
      D(9)=PH(3)
30      FORMAT(10X,3F10,3,6F10,4)
40      FORMAT(70X,3F10,4)
      RETURN
      END
      SUBROUTINE OUTPT(X,XH,P,K,TIME,II)
      REAL K

```



```

COMMON /LIM/ L
COMMON /LABLES/ LABT,LARE,LARM,LARS
DIMENSION X(9),XH(9),P(9,9),K(9,3),XXH(9)
DO 10 I=1,L
10 XXH(I)=X(I)-XH(I)
PRINT 1000,I,TIME
1000 FORMAT(/,10X,*NUMBER OF ITERATIONS USED IN FILTER = *,I5,
1 5X,*CORRESPONDING TIME = *,F10.2,/)
LARS=10H STATES
LABT=10H TRUE
LARE=10H ESTIMATED
LARM=10H - MINUS -
PRINT 2000,LART,LARS
2000 FORMAT(45X,5A10)
PRINT 3000,X
3000 FORMAT(1RX,*POSITION*,15X,15X,*VELOCITY*,15X,15X,*ACCELERATION*,/
1 3(RX,*X*,8X,8X,*Y*,8X,8X,*Z*),/
2 9(1X,E12.6,2X),/)
PRINT 2000,LARE,LARS
PRINT 3000,XH
PRINT 2000,LART,LARS,LARM,LARE,LARS
PRINT 3000,XXH
PRINT 4000,P
4000 FORMAT(/,10X,*STATE ERROR COVARIANCE MATRIX --P*,/
1 9(9(1X,E12.6,2X),/))
PRINT 5000,K
5000 FORMAT(/,10X,*GAIN MATRIX --K*,/
1 3(9(1X,E12.6,2X),/))
RETURN
END
SUBROUTINE COLLECT(X,XH,Z,II,DTIME,N)
COMMON /LABLES/ LABT,LARE,LARM,LARS
DIMENSION DXP(600),DXV(600),DXA(600),DYP(600),DYV(600),DYA(600),
1DZP(600),DZV(600),DZA(600),DXS(600),DYS(600),DZS(600)
DIMENSION X(9),XH(9),Z(3),SIG(12)
C IF PLOTS OF TRUE AND ESTIMATED STATES ARE DESIRED THEN THE
C USER SHOULD STORE ALL DESIRED DATA IN THIS ROUTINE ,FOR
C EXAMPLE XTPLOT(II)=X(1) ,XEPLLOT(II)=XH(1) , ETC...
C STORES TRUE-ESTIMATED STATES AND TRUE-SENSED STATES
DXP(II)=X(1)-XH(1)
DYP(II)=X(2)-XH(2)
DZP(II)=X(3)-XH(3)
DXV(II)=X(4)-XH(4)
DYV(II)=X(5)-XH(5)
DZV(II)=X(6)-XH(6)
DXA(II)=X(7)-XH(7)
DYA(II)=X(8)-XH(8)
DZA(II)=X(9)-XH(9)
DXS(II)=X(1)-Z(1)
DYS(II)=X(2)-Z(2)
DZS(II)=X(3)-Z(3)
RETURN
ENTRY PLOTTING
C COMPUTES MEAN SQUARE STATISTIC FOR ALL DIFFERENCES
CALL STAT1(DXP,N,SIG(1))
CALL STAT1(DYP,N,SIG(2))
CALL STAT1(DZP,N,SIG(3))
CALL STAT1(DXV,N,SIG(4))
CALL STAT1(DYV,N,SIG(5))
CALL STAT1(DZV,N,SIG(6))
CALL STAT1(DXA,N,SIG(7))
CALL STAT1(DYA,N,SIG(8))
CALL STAT1(DZA,N,SIG(9))
CALL STAT1(DXS,N,SIG(10))
CALL STAT1(DYS,N,SIG(11))
CALL STAT1(DZS,N,SIG(12))

```

```

      PRINT 1000,LAHT,LARS,LARM,LARF,LARS
1000  FORMAT(///,22X,*MEAN SQUARED ERROR ---  *5A10.//)
      PRINT 3000,(SIG(I),I=1,9)
3000  FORMAT(/,9(1X,F12.6,2X),/)
      LARSS=10*SENSOR ST.
      PRINT 1000,LAHT,LARS,LARM,LARSS,LARS
      PRINT 3000,SIG(10),SIG(11),SIG(12)
      RETURN
      END
      SUBROUTINE OWN(X,DT,NEND)
      COMMON /LIM/ L
      DIMENSION X(9),F(9,9),G(9,3),XN(3),Q(3,3),      W(3),FX(9),GW(9)
1. PG(3),TH(3),PH(3),D(9)
      SET CONSTANTS  AX,AY,AZ
      DATA AX,AY,AZ/30.,31/
      SIG=1000.
      RX=2.*AX*SIG
      RY=2.*AY*SIG
      RZ=2.*AZ*SIG
      CALL ZERO(F,L,L)
      CALL ZERO(G,L,3)
      CALL ZERO(Q,3,3)
C      SPECIFY LENGTH(NEND) AND TIME INCREMENT(DT) OF TARGET
C      FLIGHT PATH GENERATION
      DT=.1
      NEND=165
C      SPECIFY INITIAL STATE VECTOR --X
      X(1)=-254.22
      X(2)=-353.31
      X(3)=3179.49
      X(4)=160.78
      X(5)=118.78
      X(6)=-34.61
      X(7)=-1.81
      X(8)=4.63
      X(9)=-9.53
C      FLIGHT PATH ---PLANT MODEL
C      INITIALIZE TRANSITION MATRIX--F
      TEMPX=EXP(-AX*DT)
      TEMPY=EXP(-AY*DT)
      TEMpz=EXP(-AZ*DT)
      TEMAX=(1.-TEMPX)/AX
      TEMAY=(1.-TEMPY)/AY
      TEMAZ=(1.-TEMPZ)/AZ
      F(1,1)=1.
      F(2,2)=1.
      F(3,3)=1.
      F(4,4)=1.
      F(6,6)=1.
      F(7,7)=TEMPX
      F(1,4)=DT
      F(1,7)=(DT-TEMAX)/AX
      F(4,7)=TEMAX
      F(5,5)=1.
      F(2,5)=DT
      F(2,8)=(DT-TEMAX)/AY
      F(5,8)=TEMAX
      F(9,9)=TEMPZ
      F(3,6)=DT
      F(3,9)=(DT-TEMAZ)/AZ
      F(6,9)=TEMAZ
C      INITIALIZE EXCITATION MATRIX-- G
      G(1,1)=(DT*DT/2.-DT/AX +TEMAX/AX)/AX
      F(8,8)=TEMPY
      G(2,2)=(DT*DT/2.-DT/AY +TEMAX/AY)/AY
      G(3,3)=(DT*DT/2.-DT/AZ +TEMAZ/AZ)/AZ

```

```

      G (4,1)=(DT-TMAX)/AZ
      G (5,2)=(DT-TMAX)/AY
      G (6,3)=(DT-TMAX)/AZ
      G (7,1)=TMAX
      G (8,2)=TMAX
      G (9,3)=TMAX
C      INITIALIZE STATE RANDOMNESS COVARIANCE MATRIX --Q
      Q(1,1)=RX/DT
      Q(2,2)=RY/DT
      Q(3,3)=RZ/DT
      PRINT 500
500  FORMAT(30X,'THIS IS A SIMULATION FOR THE NINE STATES OF AIRCRAFT F
      ILIGHT PATH',//)
      PRINT 1000
1000  FORMAT(/,10X,'PLANT MODEL FOR AIRCRAFT FLIGHT')
      PRINT 2000
2000  FORMAT(/,10X,'DYNAMICS MODEL - F-MATRIX')
      PRINT 3000,F
3000  FORMAT(9(1X,E12.6,2X))
      PRINT 4000
4000  FORMAT(/,10X,'RANDOM MODEL - G--MATRIX')
      PRINT 3000,G
      RETURN
      ENTRY COMP
C      SIMULATE TARGET AIRCRAFT FLIGHT USING PLANT MODEL DESCRIBED
C      IN REPORT
      CALL RAND(3,XN)
      W(1)=XN(1)*SQRT(Q(1,1))
      W(2)=XN(2)*SQRT(Q(2,2))
      W(3)=XN(3)*SQRT(Q(3,3))
      CALL MPRD(F,X,FX,L,L,1)
      CALL MPRD(G,W,GW,L,3,1)
      DO 10 I=1,L
10  X(I)=FX(I)+GW(I)
      CALL RANGE(X,RG)
      CALL THET(X,TH)
      CALL PHI(X,PH)
      D(1)=RG(1)
      D(2)=RG(2)
      D(3)=RG(3)
      D(4)=TH(1)
      D(5)=TH(2)
      D(6)=TH(3)
      D(7)=PH(1)
      D(8)=PH(2)
      D(9)=PH(3)
      RETURN
      END
      SUBROUTINE STATI(D,N,SIG)
C      COMPUTES MEAN SQUARE STATISTIC
      DIMENSION D(1)
      SUM=0.
      DO 10 I=1,N
      SUM=SUM+D(I)*D(I)
10  CONTINUE
      RN=N
      SIG=SUM/RN
      RETURN
      END
      SUBROUTINE RANGE(D,X)
C      TRANSFORMS DATA IN X-Y-Z SET TO R SET
      DIMENSION D(1),X(1)
      R=SQRT( D(1)*D(1)+D(2)*D(2)+D(3)*D(3) )
      ROOT= ( D(1)*D(4)+D(2)*D(5)+D(3)*D(6) )/R
      TERM1=( D(1)*D(7)+D(2)*D(8)+D(3)*D(9) )/R
      TERM2=( D(4)*D(4)+D(5)*D(5)+D(6)*D(6) )/R

```

```

TERM3=RDDOT*RDDOT/R
RDDOT=TERM1*TERM2-TERM3
X(1)=R
X(2)=RDDOT
X(3)=RDDOT
RETURN
END
SUBROUTINE THET(D,X)
  TRANSFORMS DATA IN X-Y-Z SET TO THETA SET
  DIMENSION D(1),X(1)
  DATA 1/07,1/07
  PI=3.14159
  THETA=ATAN2(-D(1),D(2))
  IF (THETA.LE.0) THETA=THETA+2.*PI
  IF (1.EQ.0) GO TO 10
  IF (D(2).LE.0.) GO TO 10
  IF (D(1).LE.0. .AND. XOLD.GE.0.) N=N-1
  IF (D(1).GE.0. .AND. XOLD.LE.0.) N=N+1
  THETA=THETA+2.*FLOAT(N)*PI
10 X(1)=D(1)
  I=1
  R2=( D(1)*D(1)+D(2)*D(2) )
  TDDOT=( D(1)*D(5)-D(4)*D(2) )/R2
  RDDOT= D(1)*D(4)+D(2)*D(5)
  TERM1=( D(1)*D(8)-D(7)*D(2) )/R2
  TERM2=( D(1)*D(5)-D(4)*D(2) )/RDDOT*(R2*R2)
  TDDOT=TERM1-2.*TERM2
  X(1)=THETA
  X(2)=TDDOT
  X(3)=TDDOT
  RETURN
  END

```

```

      SUBROUTINE RAND(N,X)
      GENERATES RANDOM SEQUENCES OF EITHER NORMAL OR UNIFORM VARIABLES
      DIMENSION X(10),JRN(31),JSUB(31)
      IF (N) 300,10,100
10    IX = 2**24
      IY = 2**31 - 1
      XIX = IX
      JRN(1) = X(1)
      M = 31
      DO 15 K = 2,M
      IF = JRN(K-1)*591 + 13
      IF (IF) 14,14,15
14    IF = IF + IY + 1
15    JRN(K) = MOD(IF,IX)
      DO 20 J=1,M
20    JSUB(J) = J + 1
      JSUB(M) = 1
      L = JRN(M)
      I = M
      RETURN
100  DO 200 K = 1,N
      SUM = -6.0
      DO 150 K1 = 1,12
      I = JSUB(I)
      L = MOD(L + JRN(I),IX)
      JRN(I) = L
150  SUM = SUM + FLOAT(L)/XIX
200  X(K) = SUM
      RETURN
300  M = IARS(N)
      DO 400 K = 1,M
      I = JSUB(I)
      L = MOD(L + JRN(I),IX)
      JRN(I) = L
400  X(K) = FLOAT(L)/XIX
      RETURN
      END

```



```

COMMON /CORRL/ COR(3)
COR(1)=0.
COR(2)=0.
COR(3)=0.
COMMON /CORRL/ COR(3)
CALL CORR(XN)
RS=X(1)*COR(1)
THS=X(2)*COR(2)
PHS=X(3)*COR(3)
SUBROUTINE INV2(X,AA)
C      SPECIAL PURPOSE MATRIX INVERSION(2X2) ONLY USING DIRECT SOLUTION
DIMENSION X(4),AA(4)
A=X(1)
C=X(2)
R=X(3)
D=X(4)
TEST=A*D-R*C
IF(TEST.EQ. 0.) STOP
AA(1)=D/TEST
AA(2)=-C/TEST
AA(4)=A/TEST
AA(3)=-R/TEST
RETURN
END

```


DISTRIBUTION

Commander
US Army Materiel Development and
Readiness Command
5001 Eisenhower Avenue
Alexandria, VA 22333

1 Attn: DRCDE-R

1 Attn: DRCDE-W

Commander
US Army Armament Command
Rock Island, IL 61201

1 Attn: AMSAR-SA

1 Attn: AMSAR-RDT-S

1 Attn: AMSAR-RDG

1 Attn: AMSAR-RD,
Mr. Brinkman

1 Attn: DRCPM-ARGADS,
Col. R. Parker

Commander
Picatinny Arsenal
Attn: SARPA-AD-C
Dover, NY 07801

Commander
Rock Island Arsenal
Attn: SARRI-L, Dr. Beckett
Rock Island, IL 61201

Office, Secretary of Defense
The Pentagon
Attn: DDR&E, Mr. T. Kotanias
Washington, DC 20310

Commander
US Army Training & Doctrine Command
Attn: ATCD-CF-A
Ft. Monroe, VA 23651

Department of the Army
Office, Chief of Research and
Development
Washington, DC 20310

1 Attn: DARD-DDZ-C,
Dr. V. Garber

1 Attn: DARD,
Dr. M. Lasser

1 Attn: DARD-DDM-A

Office, Deputy Under Secretary
of the Army (OR)
The Pentagon
Washington, DC 20310

1 Attn: Dr. W. Payne

1 Attn: Dr. D. Willard

Commander
US Army Air Defense School
Ft. Bliss,

1 Attn: ATSA-CD

1 Attn: ATSA-CD-MS,
MAJ Emmeret

1 Attn: ATSA-CTD-MT-T

Commander
US Army Armor School
Attn: ATSB-CD-CA
Ft. Knox, KY 40121

Director
US Army TRADOC Sys Anal Act
Attn: SARWV-RDD
White Sands Msl Rg, NM 88002

Commander
Watervliet Arsenal
Attn: SARWV-RDD
Watervliet, NY 12189

DISTRIBUTION (Cont)

Project Manager
Mechanized Infantry Combat Vehicle
System
28150 Dequindre Street
Warren MI 48090

Director
Army Material System Analysis
Agency
Attn: AMXSU-D, Dr. J. Sperrazo
Aberdeen Proving Ground, MD 21005

Defense Advanced Research Projects
Agency
1400 Wilson Boulevard
Attn: Dr. E. Gerry
Arlington, VA 22209

Defense Documentation Center (12)
Cameron Station
Alexandria, VA 22314

Commander
Frankford Arsenal
Philadelphia, PA 19137

- 1 Attn: AOA-M
- 1 Attn: TD
- 1 Attn: PA
- 1 Attn: GC
- 1 Attn: FSC
- 1 Attn: FCD
- 10 Attn: FCW-D,
Mr. Walter Dziwak
- 1 Attn: MCD-A
- 3 Attn: TSP-L/51-2
1 - Circulation Copy
1 - Reference Copy

Printing & Reproduction Division
FRANKFORD ARSENAL
Date Printed: 17 Feb 1977



